K6 SERIES

kameleon™

ADAPTABILITY IN LIGHTING CONTROL

BACnet™

# MANUEL D'INSTALLATION, D'OPÉRATION ET D'ENTRETIEN




**121-38050-B**

2625, avenue Dalton, Québec, (Québec), G1P 3S9, Tél. : (418) 651-8000

kameleon[MD]

## HISTORIQUE DU DOCUMENT

| Révision | Date | Description | Auteur(s) |
|---|---|---|---|
| Émission | 2009 / 04 / 04 | Manuel d'opération du contrôleur *BACnet* d'éclairage KC601. | Bernard Vallée |
| Révision A | 2009 / 10 / 20 | Nouvelle version reflétant les changements à la console de configuration (version 1.4) avec ses nouveaux paramètres, selon l'ACI 2009-083, plus l'ajout d'une référence au PICS dans la section *BACnet*. | Bernard Vallée |
| Révision B | 2010 / 01 / 08 | Mise à jour | Bernard Vallée |

## APPROBATION DU DOCUMENT

| | | |
|---|---|---|
| **Vérifié par :** | Bernard Vallée | |
| **Vérifié par :** | Marcel Landry | |
| **Revu et approuvé par :** | Mario Lehoux | |

# TABLE DES MATIÈRES

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# 1. Introduction

Le contrôleur *BACnet* d'éclairage KC601 permet d'allumer ou d'éteindre jusqu'à 64 relais individuels pour gérer des lumières. Il permet aussi de se créer jusqu'à 16 groupes composés d'agencements des sorties/relais disponibles. Le contrôle sur l'état d'une sortie individuelle ou d'un groupe peut alors se faire à partir d'un KC401 ou d'une console communiquant selon le protocole *BACnet*, cette dernière permettant en plus de visualiser l'état du système.

## 1.1 Acronymes et abréviations

**APDU** **A**pplication layer **P**rotocol **D**ata **U**nit. L'APDU de *BACnet* consiste en un en-tête (de format fixe) indiquant entre autre la nature de la requête, le nombre d'octets identifiant l'opération de la couche d'application, des informations sur la segmentation possible du message. L'APDU de *BACnet* comprend aussi en une section variable, selon que des données accompagnent la requête.

**ASHRAE** **A**merican **S**ociety of **H**eating, **R**efrigerating and **A**ir-Conditioning **E**ngineers : Un comité de cette société, avec des représentants de tous les secteurs de l'industrie, voit à la mise à jour, au support et au maintient de la norme *BACnet*.

***BACnet*** (***B**uilding **A**utomation & **C**ontrol **Net**works)* : protocole de communication développé comme standard par *ASHRAE* pour l'automatisation et le contrôle complets des bâtiments.

**Baud Rate** Débit ou vitesse de transmission, en *« bauds »* = bits par secondes.

**Groupe** Un groupe est composé de sorties situées sur un ou plusieurs KC401 afin d'activer plusieurs sorties par une simple commande adressée au groupe; se référer à la section : *4.2 Programmation de groupes*.

**Interruption** Lorsqu'un événement survient à un port ou qu'un compteur atteint sa limite, un signal asynchrone interrompt le programme principal pour exécuter le programme associé à la source de *« l'interruption »*.

**KC401** Séquenceur de contrôle d'éclairage.

**Kameleon** Système de contrôle d'éclairage conçu par Gentec inc.

**MS/TP** Le protocole de communication **"M**aster-**S**lave/**T**oken-**P**assing" est spécifique à *BACnet* et il a été conçu spécialement pour les besoins de l'automatisation des édifices et le contrôle des appareils. Il sert à gérer les échanges (en RS485) entre les composantes d'un réseau *BACnet*. Ce protocole requiert des câbles blindés à paire torsadée (*« shielded twisted-pair cables»*).

**OFF Only** Se référer à la section : *4.4 Options pour les sorties des KC401.*

**ON Only** Se référer à la section : *4.4 Options pour les sorties des KC401.*

**PC** Ordinateur personnel *(**P**ersonal **C**omputer).*

**RS232** Protocole standard pour une ligne de communication série, *«full duplex»* (les requêtes et les réponses sont sur 2 lignes séparées). Ce protocole est utilisé pour de courtes distances, jusqu'à 50 pieds pour un *« baud rate »* de 20kb/s.

**RS485** Protocole standard pour une ligne de communication série, «*half duplex*» (les requêtes et les réponses sont sur la même ligne). Ce protocole est utilisé pour des *« baud rates »* et des distances plus élevés, jusqu'à 4000 pieds pour le *« baud rate »* de 100Kb/s, mais 40 pieds pour 10Mb/s.

**Sortie** L'un des 16 relais accessibles sur un KC401 et pouvant être inclus dans un groupe; se référer à la section : *4.3 Activation de groupes et de sorties individuelles*.

**Sortie distante** Sortie située sur un KC401 autre que celui qui émet la commande

**Timer** Compteur qui génère une *« interruption »* à la fin de son compte. Dans le contrôleur *BACnet*, le principal compteur sert à appeler le programme principal à intervalles fixes.

**TOE** Se référer à la section : *4.4 Options pour les sorties des KC401.*

**Warning** Se référer à la section : *4.4 Options pour les sorties des KC401.*

## 1.2 Références

**Documents *« Gentec »* :**

| # Gentec | Description du document |
|---|---|
| 121-35622 | Spécifications techniques du protocole V1.0 de la série K4 Kameleon. |
| 121-35701 | Manuel d'installation, d'opération et d'entretien du contrôleur d'éclairage KC401 |
| 121-37475 | Cahier de charge du projet de passerelle pour gérer quatre KC401, avec communication *BACnet.* |
| 121-37476 | Cahier de conception architecturale du projet de passerelle pour gérer quatre KC401, avec communication *BACnet.* |

**Autres documents :**

- Manuel de la norme **ANSI/ASHRAE Standard 135-2004** : *« BACnet, A Data Communication Protocol for Building Automation and Control Networks ».*

# 2. Carte du contrôleur *BACnet* d'éclairage KC601

## 2.1 Principales composantes sur la carte

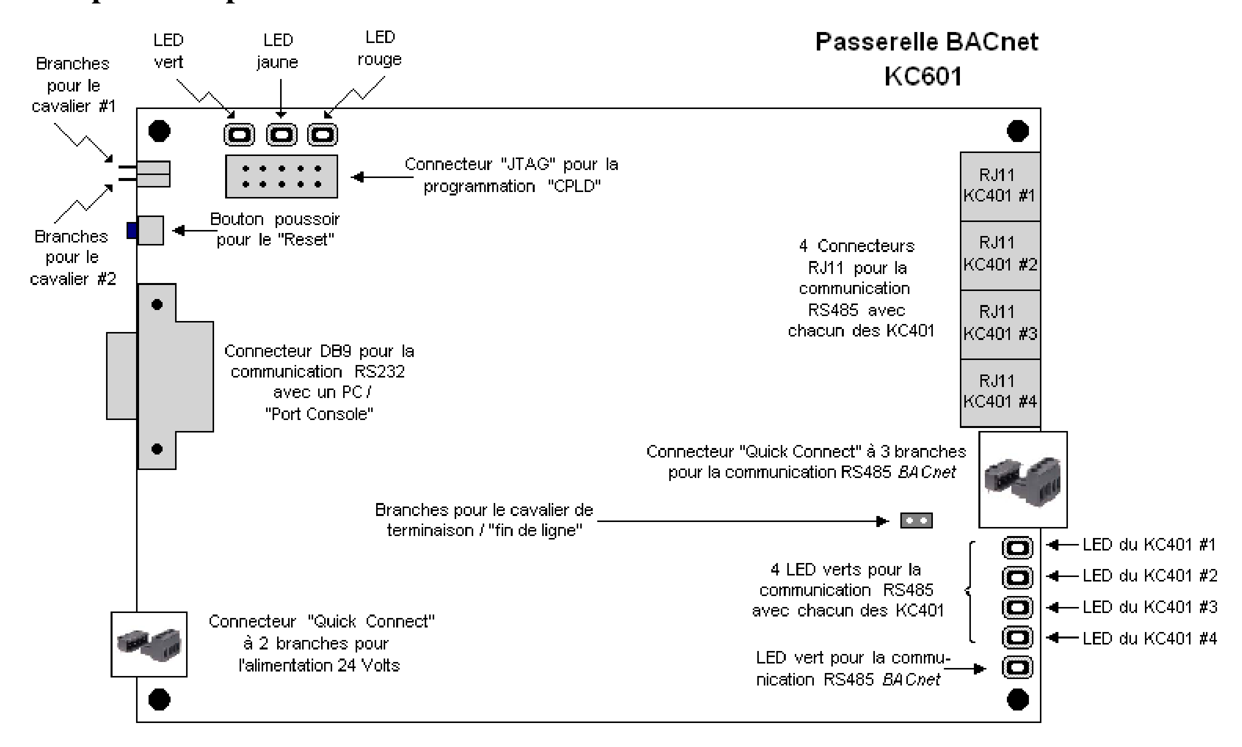



*Figure 1 : Principales composantes sur la carte KC601 pour communiquer et diagnostiquer la passerelle.*

### 2.1.1 L'alimentation

Connecteur *« Quick Connect » / « Terminal Block »* à 2 branches, fournissant 24 volts AC.

### 2.1.2 Ports de communication

a) **Connecteur DB9 pour la communication RS232 avec un PC** (voir la section *3.1.1 Sélection du port de communication avec le contrôleur BACnet d'éclairage*) **:**
Il permet de communiquer avec un ordinateur, que ce soit :
- pour échanger avec la console de configuration,
- pour recevoir une nouvelle version du programme qui gère la passerelle,
- pour transmettre des informations de diagnostique ou de débogage.

b) **Connecteur *« Quick Connect » / « Terminal Block »* à 3 branches pour la communication RS485 *BACnet* :** permet les échanges selon le protocole MS/TP avec le réseau *BACnet*.

c) **4 Connecteurs RJ11 pour la communication RS485 avec chacun des KC401 :** chacun de ces connecteurs correspond à un numéro de KC401. Il est donc important d'utiliser d'abord le connecteur pour le KC401 #1 et ainsi de suite (se référer à la *Figure 1*, section de droite) car :
- si vous n'utilisez pas quatre KC401, seules les premières sorties (en commençant par celle pour le KC401 #1) seront adressées;
- les numéros de groupes et de sorties individuelles sont automatiquement incrémentés à partir du KC401 à l'entrée #1.

d) **Connecteur *« JTAG »* pour la programmation *« CPLD »* :** ne devrait être utilisé que par un technicien Gentec.

### 2.1.3 LED / Diodes Électroluminescentes

Ils indiquent dans quel état est la carte KC601. On retrouve 2 groupes de LED :

**a) Groupe de 3 LED de couleurs différentes (Vert – Jaune – Rouge) :**

Ce groupe est situé dans le coin supérieur gauche de la *Figure 1*, tout juste derrière le connecteur *« JTAG »* pour la programmation *« CPLD »*. Les LED vert et rouge demeurent allumés s'il y a réinitialisation de la passerelle, pendant un bref moment s'il n'y a pas de cavalier et constamment dans le cas contraire.

- *LED Rouge :* il demeure allumé pour indiquer que la carte est alimentée.

  Lors d'une *réinitialisation* de la carte, il demeure aussi allumé.
- *LED Jaune :* il doit clignoter pour indiquer l'état de la configuration de la carte :
  - il clignote lentement pour indiquer qu'un *« baud rate »* a été attribué au port de communication *BACnet*;
  - il clignote rapidement pour indiquer que le port *BACnet* n'a pas de fréquence de communication *(« baud rate »)*;
  - il clignote encore plus rapidement :
    - pendant les quelques secondes que dure l'écriture d'une configuration en mémoire flash, soit après la transmission d'une nouvelle configuration par la console Windows, soit au retour au mode *« RUN »* après avoir changé la composition d'un ou plusieurs groupes à partir des KC401 en mode *« PROGRAM »* (voir la section : *4.2 Programmation de groupes*);
    - en permanence si un problème est survenu au cours de l'écriture dans la mémoire flash.

  Lors d'une *réinitialisation* de la carte, le LED jaune demeure éteint.
- *LED Vert :* aussi appelé *« LED du port console »*, il s'allume pour chaque octet échangé sur le port de communication RS232 relié à un PC.

  Lors d'une *réinitialisation* de la carte, il demeure faiblement allumé.

**b) Groupe de 5 LED de couleur verte :**

Ce groupe est situé dans le coin inférieur droit de la *Figure 1*.

- *4 LED pour les KC401 :* chacun de ces LED est associé à un port de communication RS485 pour un KC401. Le LED indique s'il y a des échanges avec le KC401 auquel le LED est associé. À noter qu'un seul KC401 à la fois est interrogé par la passerelle, en commençant par le KC401 #1.

  Lors d'une *réinitialisation* de la carte, ces 4 LED demeurent allumés.
- *LED pour la communication BACnet :* il est associé au port de communication RS485 *BACnet* et ne clignotera donc pas s'il n'y a pas de *« baud rate »* d'attribué au port correspondant.

Lors d'une *réinitialisation* de la carte, ce LED demeure faiblement allumé si le cavalier #2 n'est pas connecté, mais éclaire autant que les 4 autres du groupe si le cavalier #2 est branché.

#### 2.1.4 Bouton poussoir pour le *« Reset »*

Il permet une réinitialisation « à froid » *(« cold reboot » / (« cold start »)* de la passerelle :

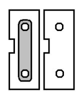
- Toutes les variables sont remises à leurs valeurs de départ, les compteurs pour la communication étant remis à zéro;
- Tous les processus de communication, incluant les requêtes *BACnet* en attente, sont réinitialisés.
- Tous les compteurs pour les diagnostiques sont réinitialisés
- À noter qu'il ne s'agit pas d'une réinitialisation des valeurs par défaut établies à l'usine.
- La composition des groupes et les paramètres configurés sont donc préservés.

#### 2.1.5 Branches pour les cavaliers *(« jumpers »)* :

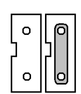
**a) Branches pour les 2 cavaliers de programmation :**

Situés dans le coin supérieur gauche de la *Figure 1*, ils permettent de faire démarrer la carte en mode programmation après avoir réinitialisé la carte. Tous les LED restent alors figés dans l'état de réinitialisation (se référer à la section : *2.1.3 LED / Diodes Électroluminescentes*). Un seul de ces 2 cavaliers doit être branché à la fois :

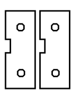
**Cavaliers**
**1 2**



**Cavalier 1 branché :** Pour transmettre un nouveau programme « usager », de plus haut niveau, qui sert à contrôler le fonctionnement de la passerelle.



**Cavalier 2 branché :** Pour transmettre un nouveau programme *« loader »*, plus rudimentaire avec les fonctions de base qui permettent entre autre de recevoir ou de changer le programme « usager ».



**Aucun cavalier :** la passerelle fonctionnera normalement sans « geler » après une réinitialisation.

**b) Branches pour le cavalier de terminaison / fin de ligne :**

Situés dans la section de droite, vers le bas, de la *Figure 1*, il permet de « fermer une ligne » de communication. Ce cavalier ne doit donc être installé que si aucun *« BACnet device »* n'est installé après le KC601 concerné, sur la même ligne de communication *BACnet*.

## 2.2 Raccords à la carte du contrôleur KC601

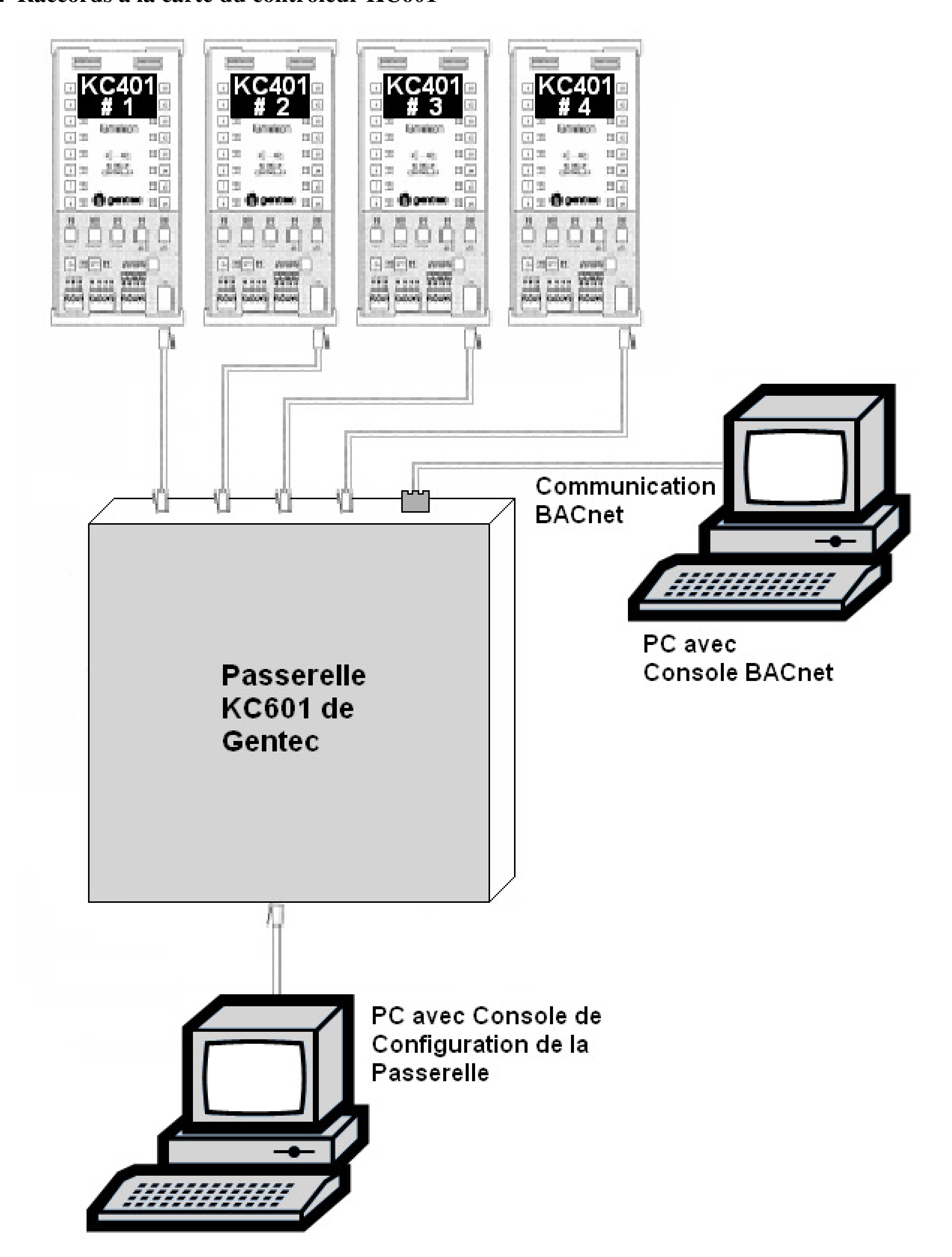



*Figure 2 : Branchements des diverses composantes reliées au contrôleur BACnet d'éclairage KC601.*

- **Console de configuration du contrôleur *BACnet* d'éclairage KC601 :** *(se référer à la section : 3. Console de configuration du contrôleur BACnet d'éclairage)*
  - Développée chez Gentec, elle est surtout requise pour la configuration initiale du KC601 ou si on veut changer le mode de fonctionnement de la passerelle KC601. Un ordinateur n'a donc pas besoin d'être relié continuellement au contrôleur *BACnet* à cette fin.
  - Un port COM (communication RS232) de l'ordinateur doit être relié au port DB9 RS232 de la carte KC601 avec un câble *« pin to pin »* (par opposition à *« null modem »*).

- **KC401 :** *(se référer à la section : 4. Contrôleurs de relais KC401 / Programmation des groupes)*
  - Chaque relais servant à contrôler une source de lumière doit être branché à l'une des 16 sorties individuelles d'un KC401.
  - Entre 1 et 4 KC401 peuvent être connectés au contrôleur *BACnet* par des câbles pour la communication RS485, avec des connecteurs RJ11 aux 2 extrémités.
  - Le contrôleur *BACnet* d'éclairage identifie le numéro d'un KC401 selon le port auquel il est branché. On doit donc utiliser d'abord le port le plus à gauche sur la *Figure 2*, puis le suivant… Se référer aussi à la section de droite de la *Figure 1*.

- **Console *BACnet* :** *(se référer à la section* : *5. La carte de contrôle KC601 en tant qu'objet BACnet)*
  - Différentes compagnies ont développé une console communiquant selon le protocole *BACnet*. Cette console permet entre autre :
    - de visualiser l'état du réseau BACnet, dont celui des KC401 reliés au KC601, et de changer manuellement l'état d'un groupe de sorties ou d'une sortie individuelle;
    - de créer des horodateurs (appelés familièrement *« céduleurs »*) afin de changer l'état des sorties automatiquement, selon un horaire établi;
    - d'envoyer aux KC601 des requêtes de réinitialisation (*« cold reboot »/« cold start »* ou *« warm reboot »/ « warm start »*) et des requêtes de désactivation pour un certain nombre de minutes, ce délai étant transmis avec la requête; se référer à la section : *5.2 Commandes/Services BACnet supportés (selon* les *modèle BIBBs / BACnet Interoperability Building Blocks – Annexe K de la norme ASHRAE)*
  - Pour être visibles à une console *BACnet*, un KC601 doit être relié au réseau via son port RS485 *BACnet* (voir la subdivision inférieure droite de la *Figure 1*), la section du réseau sur laquelle le KC601 se retrouve étant gérée habituellement par un serveur ou une passerelle *BACnet*.
  - La console *BACnet* peut être installée sur le même ordinateur que la console de configuration.

# 3. Console de configuration du contrôleur *BACnet* d'éclairage

Lors du premier démarrage de la console de configuration *BACnet*, une fenêtre vous permettra de choisir la langue d'affichage pour cette console de configuration :




*Figure 3 : Fenêtre pour choisir la langue d'affichage de la console.*

Dès que vous aurez coché la langue qui sera utilisée pour l'affichage de la console, le bouton **[ OK ]** deviendra accessible. Après avoir cliqué sur ce bouton, la fenêtre principale de la console apparaîtra. Par la suite, chaque fois que vous redémarrerez la console, la fenêtre du choix de la langue ne s'affichera plus; la console vous présentera dès le départ la fenêtre principale dans la langue que vous avez choisie lors de son premier démarrage.

**NOTE :**

Si vous voulez changer la langue d'affichage, vous n'aurez qu'à détruire le fichier : *« Language.bin »* dans le répertoire où est installée la console. Lors du prochain démarrage de la console, vous verrez d'abord la fenêtre permettant de choisir la langue d'affichage.

La console se subdivise en 2 fenêtres accessibles via des onglets :

- Onglet **« Configuration » :** c'est la fenêtre qui apparaît lors du démarrage de la console. Elle permet de configurer le mode de fonctionnement du KC601.
- Onglet **« Communication » :** cette fenêtre nous permet d'obtenir les statistiques tenues dans le KC601 sur la communication *BACnet* telle que perçue par le port RS485 (du KC601) affecté à cette fin.

### 3.1 Onglet « *Configuration* » / Fenêtre principale de la console de configuration

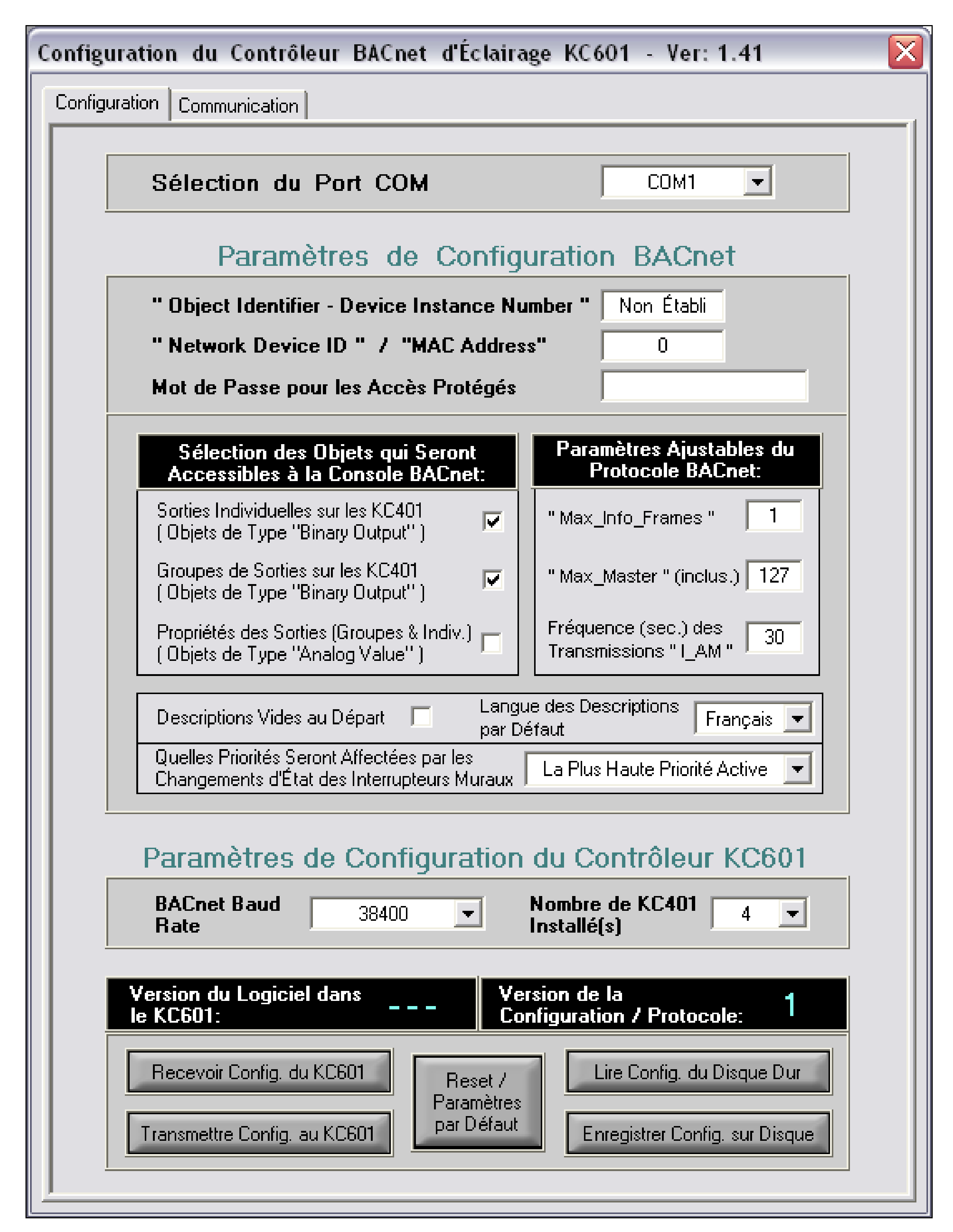



*Figure 4 : Fenêtre principale la console de configuration du contrôleur BACnet d'éclairage KC601.*

### 3.1.1 Sélection du port de communication avec le contrôleur *BACnet* d'éclairage




- On indique ici quel port de l'ordinateur est utilisé pour le relier à l'entrée DB9/RS232 de la carte KC601 (à gauche de la *Figure 1*). Ce port est configuré comme suit :
  - 8 bits de données
  - 1 bit d'arrêt
  - Parité = NONE
  - Type de données = Binaires
  - Baud Rate = 19200.
- Au démarrage, la console recherche le premier port COM disponible, en partant de *« COM1 »*. Si aucun port n'est disponible, la console affichera *« None »* pour ce champ.
- En cliquant sur la flèche qui pointe vers le bas, une liste vous permettra de choisir parmi tous les ports COM détectés comme étant disponibles, entre COM1 et COM20 inclusivement. La liste débute par *« None »* afin de vous permettre de ne pas activer de port COM.

**ATTENTION :**

Pour la communication RS232, si vous utilisez un convertisseur USB (particulièrement avec un ordinateur portable), il arrive fréquemment que Windows choisisse par défaut un mauvais pilote (*« driver »*). Il ne suffit pas alors de choisir le bon pilote; il faut utiliser un logiciel pour éliminer la sélection faite par Windows avant d'installer le pilote requis pour votre convertisseur.

### 3.1.2 Paramètres de configuration *BACnet*

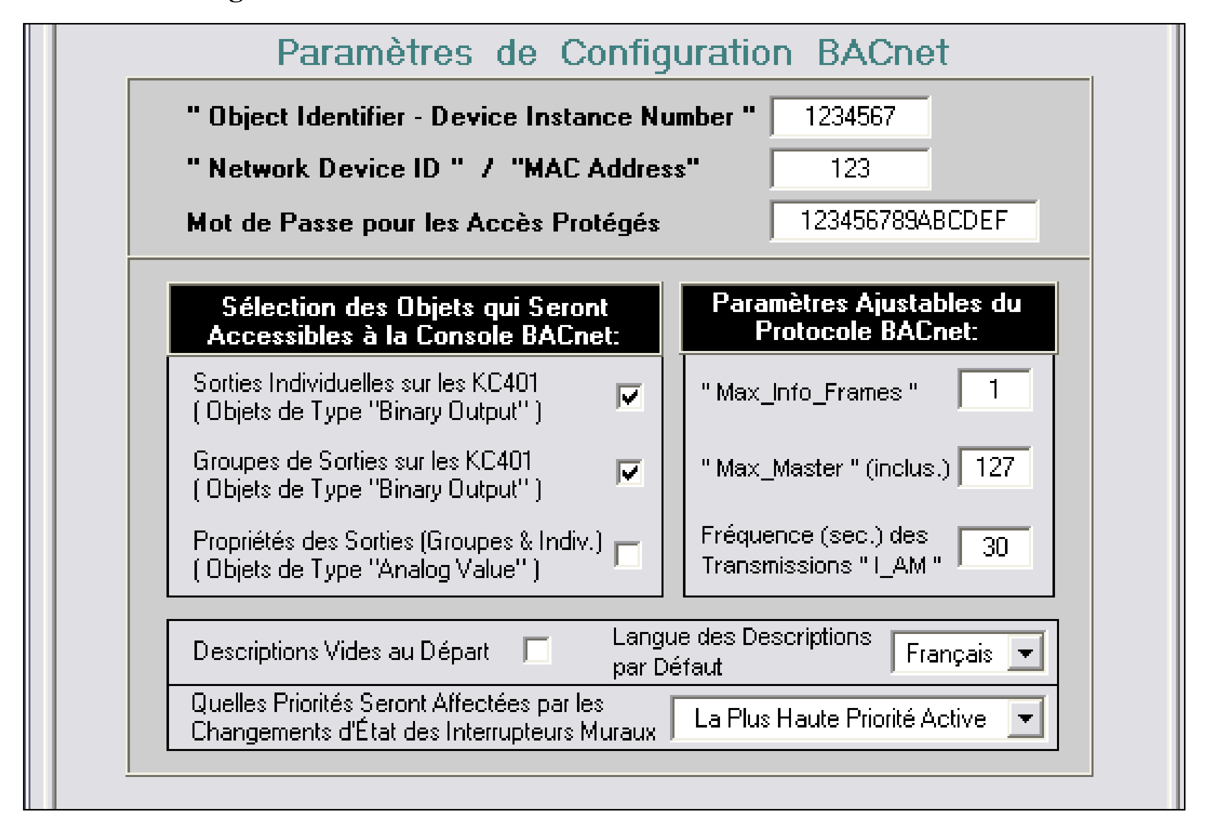



*Figure 5 : Fenêtre principale – Paramètres de configuration BACnet du KC601.*

Ce sont les paramètres qui relèvent de l'environnement *BACnet*, afin de définir le KC601 en tant qu'objet *BACnet* / *« BACnet Device »* et :

- d'établir quels sont les objets *BACnet* qui composent le KC601 et qui seront accessibles par une console *BACnet*;
- de déterminer comment certaines informations seront retournées à cette console *BACnet*;
- de préciser certains ajustements sur le protocole de communication *BACnet.*

### 3.1.2.1 Paramètres de base pour établir le KC601 en tant qu'objet *BACnet*

" Object Identifier - Device Instance Number " 1234567

C'est le **numéro d'identification** qui caractérise le KC601 en tant qu'objet *BACnet* unique. Il agit un peu comme un numéro de série qui permet de différencier un objet *BACnet* de tous les autres. Ce numéro doit donc être unique parmi tous les objets *BACnet* de type *« Device »* installés sur le réseau. À noter qu'il ne s'agit pas d'une adresse réseau et que ce *« BACnet Object Instance Number »* sera utilisé par une console *BACnet* pour découvrir son adresse réseau. Les valeurs permises sont entre 0 et 0x3FFFFF (= 4194303), cette dernière valeur étant exclue mais étant utilisée comme valeur par défaut pour indiquer que le paramètre est *« Non Établi »*.




C'est l'adresse qui identifie le contrôleur *BACnet* d'éclairage sur le « réseau de communication » *BACnet.* L'adresse 0xFF (« 255 ») est réservée pour le *« broadcast »*, i.e. pour adresser toutes les composantes simultanément sans en viser une en particulier. Il est recommandé pour certains types de réseaux de ne pas dépasser l'adresse 127 en MS/TP. Ainsi, comme le KC601 agit comme un *« nœud maître » / « master node »* sur un réseau MS/TP, on ne doit pas lui donner une adresse au-dessus de 127, les adresses de 128 à 254 étant réservées pour les *« nœuds esclaves » / « slave nodes »*.

**NOTE :** Si la valeur entrée ici est plus grande que celle du paramètre *« Max_Master »*, ce dernier paramètre sera ajusté à la nouvelle valeur du *« Network Device ID »*.

Mot de Passe pour les Accès Protégés 123456789ABCDEF

Certaines commandes *BACnet* sont sécurisées par un mot de passe, comme les requêtes de réinitialisation ou pour désactiver un appareil *BACnet* pendant une période de temps. La console permet que le mot de passe soit composé d'un maximum de 15 caractères alphanumériques, les lettres étant converties en majuscules. Si vous ne voulez pas de mot de passe, vous n'avez qu'à laisser ce champ vide.

#### 3.1.2.2 Sélections des objets *BACnet* du KC601 accessibles via une console *BACnet*

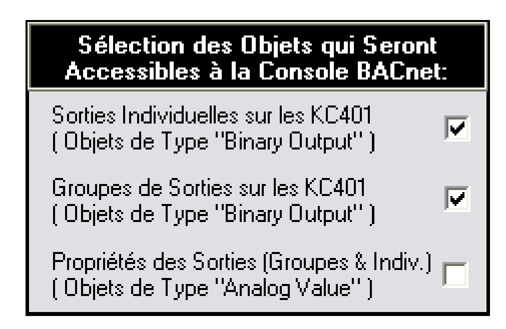
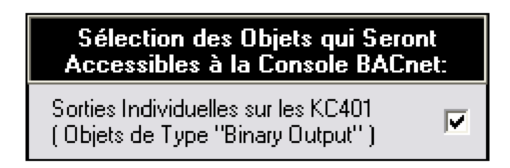



*Figure 6 : Fenêtre principale – Sélection des objets du KC601 disponibles à partir d'une console BACnet.*

Il s'agit de 3 cases à cocher qui permettent de choisir parmi tous les objets *BACnet* disponibles sur le KC601, lesquels seront affichés par une console *BACnet*.




En cochant cette case, pour chaque KC401 contrôlé par le KC601, la console *BACnet* affichera 16 objets *BACnet* de type *« Binary Output »* pour représenter chacune des 16 **sorties individuelles** que l'on retrouve sur un KC401, de sorte que vous pourrez :

- surveiller quel est l'état courant de chacune des sorties individuelles;
- changer l'état de chacune des sorties individuelles à partir de la console, que ce soit manuellement ou via un horodateur.

Ainsi, si le KC601 contrôle quatre KC401, la console *BACnet* affichera 64 objets *BACnet* de type *« Binary Output »* juste pour refléter les sorties individuelles disponibles sur les KC401 reliés au KC601.

Se référer à la section : *5.1.2 « Binary Output Object Type »* pour plus de détails sur ce type d'objets *BACnet*.




En cochant cette case, pour chaque KC401 contrôlé par le KC601, la console *BACnet* affichera 4 objets *BACnet* de type *« Binary Output »* pour représenter chacun des 4 **groupes de sorties** que l'on retrouve sur un KC401.

Ainsi, si le KC601 contrôle quatre KC401, la console *BACnet* affichera 16 objets *BACnet* de type *« Binary Output »* juste pour refléter les groupes de sorties disponibles sur les KC401 reliés au KC601. Si en plus la case *« Sorties Individuelles sur les KC401 »* était aussi cochée, pour quatre KC401, il y aurait en tout 80 (➔ 16 + 64) objets de type *« Binary Output »* pour représenter tous les groupes et toutes les sorties individuelles disponible via le KC601.

Se référer à la section : *5.1.2 « Binary Output Object Type »* pour plus de détails sur ce type d'objets *BACnet*.

**NOTE :**

Le KC601 communique via *BACnet* afin permettre le contrôle de ses sorties, individuelles ou groupes, à partir d'une console *BACnet*. Vous devez donc cocher au minimum une des 2 cases à cet effet. D'ailleurs, la console de configuration ne vous permet pas de ne pas choisir d'objet de type *« Binary Output »*. Ainsi, si vous décochez par exemple la case *« Sorties Individuelles sur les KC401 »* alors que la case pour afficher des *« Groupes de Sorties sur les KC401 »* est déjà décochée, le configurateur sélectionnera automatiquement la seconde. L'inverse se confirmerait aussi.

Propriétés des Sorties (Groupes & Indiv.)
( Objets de Type "Analog Value" ) ☐

Cette option permet de visualiser à partir d'une console *BACnet* des objets de type *« Analog Value »* pour chaque sortie disponible :

- les premiers objets *BACnet* de type *« Analog Value »* sont associés aux groupes, si la case *« Groupes de Sorties sur les KC401 »* a été cochée,
- les objets *« Analog Value »* suivants sont associés aux sorties individuelles, si la case *« Sorties Individuelles sur les KC401 »* a été cochée.

Les valeurs analogiques affichées correspondent à un champ de bits, les valeurs affichées n'étant que des « 0 » et des « 1 », « 0 » indiquant que l'option correspondant à la position où se retrouve cette valeur est désactivée… La description qui accompagne chacune des valeurs analogiques décrit à quoi consiste chacune des valeurs; pour une configuration à un seul KC401 pour lequel on fait afficher tous les types d'objets BACnet disponibles, la description affichée par une console *BACnet* pour les objets de type « Analog Values » ressemblerait à :

| Description |
|---|
| Groupe_01= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| Groupe_02= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| Groupe_03= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| Groupe_04= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| Sortie No.01 = 10:TOE, 01:WARNING |
| Sortie No.02 = 10:TOE, 01:WARNING |
| Sortie No.03 = 10:TOE, 01:WARNING |
| Sortie No.04 = 10:TOE, 01:WARNING |
| Sortie No.05 = 10:TOE, 01:WARNING |
| Sortie No.06 = 10:TOE, 01:WARNING |
| Sortie No.07 = 10:TOE, 01:WARNING |
| Sortie No.08 = 10:TOE, 01:WARNING |
| Sortie No.09 = 10:TOE, 01:WARNING |
| Sortie No.10 = 10:TOE, 01:WARNING |
| Sortie No.11 = 10:TOE, 01:WARNING |
| Sortie No.12 = 10:TOE, 01:WARNING |
| Sortie No.13 = 10:TOE, 01:WARNING |
| Sortie No.14 = 10:TOE, 01:WARNING |
| Sortie No.15 = 10:TOE, 01:WARNING |
| Sortie No.16 = 10:TOE, 01:WARNING |

*Figure 7 : Console BACnet : Affichage des descriptions pour les objets de type « Analog Values » du KC601.*

Le contrôleur KC601 ne fait que transmettre les états courants et les descriptions pour ces *« valeurs analogiques »* sans possibilité de les modifier à partir d'une console *BACnet*. Se référer à la section : *5.1.3 « Analog Value Object Type »*.

#### 3.1.2.3 Paramètres ajustables du protocole *BACnet*

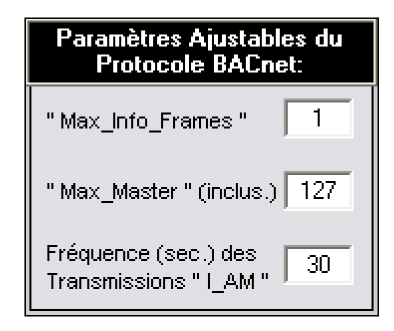
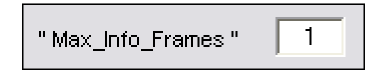
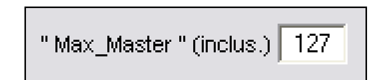



*Figure 8 : Fenêtre principale - Paramètres ajustables du protocole de communication BACnet.*

Ces 3 paramètres permettent d'optimiser/restreindre les échanges entre le KC601 et le reste du réseau *BACnet*.




Le KC601 agit en *« nœud maître »* dans un réseau de communication *MS/TP*. La propriété : *« Max_Info_Frames »* permet de spécifier le nombre maximum de *« trames d'information »* que le nœud peut transmettre avant de devoir passer le *« jeton » (« token »)* à la station suivante.




Le KC601 agissant en *« nœud maître »* dans un réseau de communication *MS/TP,* lorsqu'il reçoit le jeton *(« token »),* avant de le refiler au *« nœud maître »* suivant :

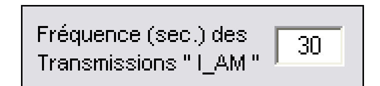
- le KC601 peut ne pas l'utiliser, i.e. n'effectuer aucune transmission avant de le refiler immédiatement au *« nœud maître »* suivant ;
- transmettre une réponse en attente (s'il avait reçu auparavant une requête *BACnet* qui s'adressait à lui et qui requiert une réponse) ;
- démarrer une nouvelle séquence d'entretien du réseau via des requêtes *« Maintenance PFM (Poll For Master) »* afin de découvrir si un nouveau *« nœud maître »* s'est inséré à une adresse réseau se situant entre celle du KC601 et l'adresse du *« nœud maître »* suivant.
  - Le KC601 démarre alors le processus de recherche à partir de sa propre adresse réseau plus un, par exemple si le KC601 a pour *« network ID »* l'adresse 100, il débutera ses requêtes *« Maintenance PFM »* à l'adresse 101.
  - À chaque requête de recherche suivante, l'adresse sera incrémentée de un jusqu'à ce que le KC601 reçoive une réponse (lui indiquant quelle est l'adresse du *« nœud maître »* suivant) ou qu'il ait atteint sa propre adresse, ce qui lui indiquerait qu'il est la seule composante sur son réseau.
  - Au cours du processus de recherche, si le KC601 a atteint l'adresse réseau la plus élevée, soit l'adresse *« Max_Master + 1 »* (*« Max_Master »* étant à 127 dans notre exemple), au lieu d'interroger l'adresse 128, il poursuivra sa recherche à l'adresse zéro et ainsi de suite jusqu'à ce qu'un *« nœud maître »* lui réponde et devienne ainsi l'interlocuteur auquel le KC601 refilera le *« token »* chaque fois qu'il aura terminé de s'en servir.

L'utilité du paramètre *« Max_Master »* est de « raccourcir/optimiser » le temps passé à rechercher le prochain nœud maître de sorte que pour un KC601 qui se retrouve avec l'adresse la plus élevée sur sa portion du réseau *BACnet*, à l'adresse 100 si on poursuit notre exemple, en établissant *« Max_Master »* à 101, notre KC601 sautera à zéro au lieu de questionner l'adresse 102 et il ne perdra pas son temps à rechercher entre 102 et 127, le KC601 devant alors attendre à chaque fois le délai d'expiration pour recevoir une réponse…

**Attention :** En utilisant toujours le même exemple, si vous ajoutez une composante *BACnet* dans le réseau à l'adresse 110, il ne sera pas découvert par les *« nœuds maîtres »* déjà installés puisqu'il ne sera pas interrogé lors de requêtes pour la gestion du réseau : *« Maintenance PFM »…*

**Valeurs permises par la console pour *« Max_Master »* :**

- Si vous essayez d'entrer une valeur inférieure à celle définie pour le paramètre : *« Network Device ID »*, la console de configuration rajustera automatiquement la nouvelle valeur entrée pour *« Max_Master »* à celle déjà établie pour *« Network Device ID »*.
- Pour ce qui est des valeurs permises pour le paramètre *« Network Device ID »*, se référer à la section : *3.1.2.1 Paramètres de base pour établir le KC601 en tant qu'objet BACnet.*




Le contrôleur KC601 transmet des messages *BACnet « I_AM »* :

- au démarrage du KC601 (au moment d'être alimenté ou après un redémarrage *« Cold Restart »* obtenu soit en appuyant sur le bouton *« Reset »* à même la carte KC601, soit par une requête *BACnet « Cold Restart »*) ;
- via une requête BACnet *« Who_Is »* (si cette requête vise une étendue de *« BACnet Object Instance Numbers »,* le KC601 doit posséder un numéro d'identification qui se situe à l'intérieur de l'étendue recherchée) ;
- à la fréquence indiquée dans la case du paramètre *« Fréquence des Transmissions I_AM »*. Pour notre exemple, le KC601 ainsi configuré transmettra ses coordonnées via un message *« I_AM »* chaque fois qu'il recevra le jeton et que 30 secondes se seront écoulées depuis la dernière émission du même message.
  - Une valeur à zéro pour ce paramètre empêche la transmission cyclique du message *« I_AM »* et optimise ainsi la communication *BACnet*.
  - La transmission cyclique du message *« I_AM »* assure la détection automatique du KC601 sur le réseau sans avoir à le réinitialiser ou à démarrer le processus de recherche via la console *BACnet*.

#### 3.1.2.4 Autres paramètres ajustables pour les échanges *BACnet*




*Figure 9 : Fenêtre principale – Autres paramètres ajustables pour les échanges BACnet.*




Exception faite de l'objet *BACnet « Device »* qui représente la carte KC601 dans son ensemble dans une console BACnet, tous les autres objets *BACnet* que l'on retrouve sur le KC601 (de types *« Binary Output »* et *« Analog value »*) sont définis avec une description exhaustive, pouvant contenir jusqu'à une cinquantaine de caractères. En cochant cette case, le KC601 ne transmettra que des champs nuls pour cette propriété, peu importe l'objet *BACnet* interrogé.




Vous choisissez ici dans quelle langue (Anglais ou Français) seront affichées les descriptions retournées par le KC601, si la case précédente a été cochée.

- o Cette option a été séparée de la possibilité de ne pas afficher de description parce qu'elle permet aussi de choisir dans quelle langue seront retournées d'autres informations comme le nom attribué à l'objet *BACnet : « device »* qui représente le KC601, de même que sa localisation (*« location »*).
- o Ne pas confondre le choix de la langue des descriptions retournées par le KC601 avec le choix de la langue d'affichage de la console de configuration…




Selon le choix fait des objets du KC601 qui seront représentés sur une console *BACnet* (voir la *Figure 6*), toutes les sorties individuelles et tous les groupes de sorties disponibles sur le KC601 peuvent être représentés dans l'environnement *BACnet* par un objet *BACnet* de type *« Binary Output »*. La norme *ASHRAE* exige que l'on puisse changer l'état (*« Present Value »*) de ce type d'objet en lui associant un niveau de priorité entre 1 et 16 inclusivement ; se référer à *l'Annexe 1. Tables des priorités pour les sorties binaires* pour plus de détails sur le fonctionnement d'une telle table.

Habituellement, il y a un interrupteur mural associé à chacune des sorties individuelles utilisées et il en va de même pour chacun des groupes de sorties configurés. Le KC601 n'associe pas de priorité aux requêtes *« ON / OFF »* transmises par un interrupteur. En contrepartie, vous pouvez établir ici comment un changement d'état issu d'un interrupteur (ou d'un bouton sur le KC601) va agir sur les tables de priorités :

- o 

**:**
  - **pour un interrupteur associé à une sortie individuelle :** si la table de priorités de la sortie binaire associée à l'interrupteur contient plusieurs priorités actives (voir les exemples de l'*Annexe 1.a Les tables de priorité selon la norme ASHRAE*), le nouvel état de l'interrupteur s'appliquera uniquement à la priorité la plus élevée;

- **pour un interrupteur associé à un groupe de sorties :** chacune des sorties individuelles du groupe étant associée elle-même à une « sortie binaire », l'application du nouvel état du groupe se reflétera à la priorité la plus élevée pour chacune des sorties individuelles appartenant au groupe :
  - **Exemple :** pour une requête de priorité 5 faite à une groupe, pour chacune des sorties individuelles du groupe, ce n'est pas la priorité 5 qui héritera nécessairement du nouvel état du groupe, mais la priorité qui avait un état non nul le plus élevé.
  - Si une sortie individuelle avait toutes les priorités de sa table avec des états nuls (➔ aucune priorité effective), alors le nouvel état du groupe s'appliquera à la sortie sans être associé à une priorité.
  - Ainsi, toutes les sorties individuelles d'un groupe risquent de ne pas toutes être à la même priorité. Avec l'option *« La Plus Haute Priorité Active »,* le nouvel état du groupe s'appliquera alors à diverses priorités pour les sorties individuelles du groupe, en autant qu'il s'agisse de la priorité la plus élevée pour chacune des sorties visées.

o 

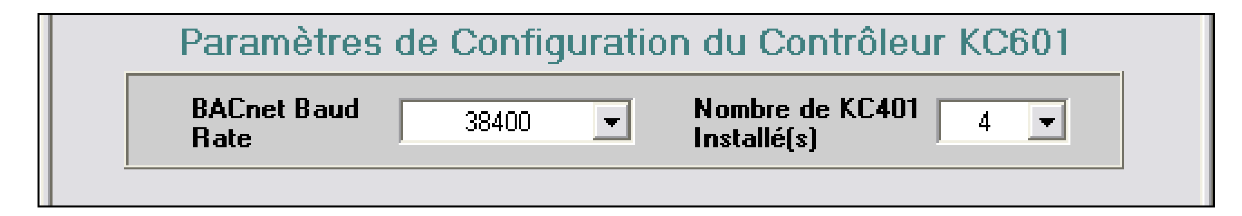
**:**

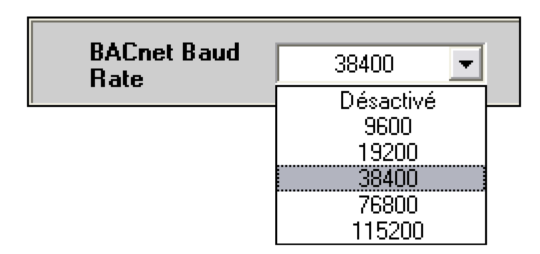
- **pour un interrupteur associé à une sortie individuelle :** toutes les priorités avec un état non nul prendront l'état de l'interrupteur.
- **pour un interrupteur associé à un groupe de sorties :** le même principe s'applique pour la table de priorités du groupe associé à l'interrupteur mais en plus, les tables de priorités de chacune des sorties individuelles verront-elles aussi les états non nuls de leurs tables de priorités être remplacés par l'état de l'interrupteur.

**Autres règles s'appliquant lors d'un changement d'état d'un interrupteur, peu importe l'option choisie pour le paramètre de configuration correspondant :**

- Si le changement d'état de l'interrupteur s'applique à une sortie binaire qui n'a aucune priorité effective, le nouvel état s'appliquera quand même à la sortie et ce nouvel état sera retourné à la console *BACnet* sans être associé à une priorité.
- Dans tous les cas, un changement d'état demandé par un interrupteur s'applique immédiatement sur le relais (à moins d'un bris ou qu'une option programmée avec un groupe ne s'applique et empêche le changement d'état demandé par l'interrupteur ; se référer à la section *4.4 Options pour les sorties des KC401*).
- Si la dernière priorité effective d'un groupe est libérée par une requête *BACnet*, la valeur par défaut établie pour : *« Relinquish_Default »* est appliquée, donc toutes les sorties seront mises à *« OFF »* avec le KC601. Les tables de priorités des sorties individuelles concernées seront mises à jour en respectant le fonctionnement de l'option choisie pour le paramètre : *« Quelles priorités Seront Affectées par les Changements d'États des Interrupteurs Muraux »*.

### 3.1.3 Paramètres de configuration du contrôleur KC601

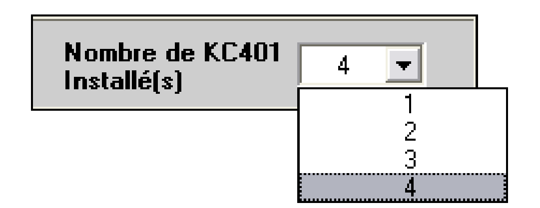



**<u>Port RS485 dédié à BACnet</u> :**

On établit à quelle vitesse va communiquer le port de communication RS485 (sur la carte KC601) dédié à *BACnet*.

- **Désactivé :** La console permet de choisir la désactivation du *« baud rate »*, mais vous ne pourrez transmettre cette valeur au contrôleur *BACnet* d'éclairage. En fait, cette option n'est là que parce qu'au départ, un nouveau contrôleur *BACnet* qui n'a jamais été configurée est dans cet état.
- **4 valeurs requises par la norme *ASHRAE* :** 9600, 19200, 38400 (valeur par défaut du configurateur) et 76800.
- **Vitesse supplémentaire :** 115200.

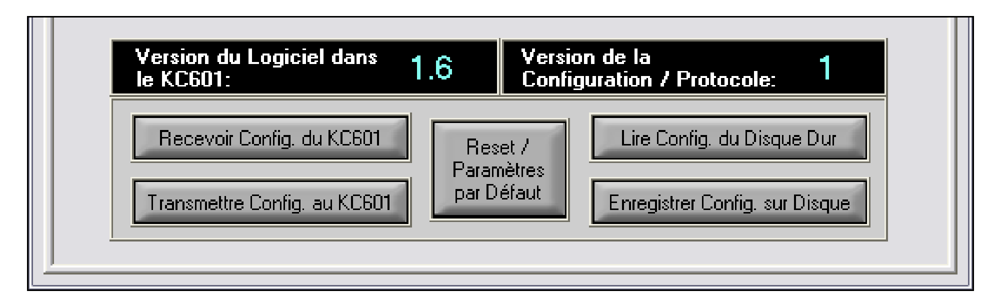



**<u>4 ports RS485 dédiés aux KC401</u> :**

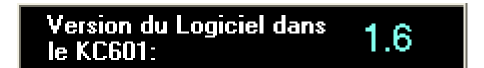
On établit quels sont les ports consécutifs dédiés aux KC401 qui seront utilisés en indiquant le nombre de KC401 qui seront interrogés par le contrôleur KC601.

- Le port assigné au KC401 #1 sera toujours utilisé; se référer à la section de droite de la *Figure 1 : Principales composantes sur la carte KC601 pour communiquer et diagnostiquer la passerelle.*
- Si vous choisissez une **valeur inférieure au nombre de KC401** reliés au contrôleur KC601, le ou les KC401 au-delà de la limite établie ne seront jamais interrogés ni concernés lors de la création de groupes. De même, si par exemple vous aviez initialement établi le nombre de KC401 à quatre pour ensuite créer des groupes incluant les quatre KC401 et que vous établissez ensuite le nombre de KC401 interrogés à deux, les derniers KC401 ne seront pas adressés lorsque viendra le temps de modifier l'état du groupe.
- Si vous choisissez une **valeur supérieure au nombre de KC401** reliés au contrôleur KC601, le contrôleur tentera de communiquer avec des KC401 inexistants, ce qui ralentira la communication, en plus de retourner à la console *BACnet* des informations sur les sorties inexistantes, en indiquant que ces sorties sont en faute puisque le KC601 les interrogera sans recevoir de réponse.

### 3.1.4 Section des boutons de contrôle et d'affichage des versions :




#### 3.1.4.1 Affichage des versions




- Au démarrage de la console ou après avoir cliqué sur le bouton *« Reset / Paramètres par Défaut »*, il n'y a aucune version d'indiquée, ce qui est signalé par 3 tirets : « - - - ».
- Une version est affichée uniquement lorsque le KC601 est interrogé via le bouton : *« Recevoir Config. Du KC601 »* et qu'il retourne une configuration valide.




- Au démarrage de la console, la valeur indiquée est déjà à *« 1 »* car il s'agit du format courant, i.e. le format de configuration qui sera transmis vers le KC601 (via le bouton : *« Transmettre Config. au KC601 »*) ou sauvegardé sur une unité de stockage (via le bouton : *« Enregistrer Config. sur Disque »*).
- Actuellement, le seul format accepté et reconnu est le format courant, soit la version 1.

#### 3.1.4.2 Échange de configuration avec le KC601




- Pour vérifier la configuration actuellement en cours dans le KC601. À partir du moment où ce bouton est pressé, les champs et les boutons de la fenêtre de configuration demeurent bloqués tant que la requête n'est pas complétée, avec ou sans succès.
  - Si le processus se déroule bien, la fenêtre sera mise à jour avec les paramètres reçus.
  - Si vous n'avez pas de réponse du KC601, un message vous en informera au bout de quelques secondes.
  - De même, si une réponse est reçue mais qu'elle ne corresponde pas au format requis, un message vous signalera pourquoi la réponse est rejetée.
- **Si vous éprouvez des problèmes de communication avec le KC601 :**
  - Vérifiez que le bon port de communication a été choisi et qu'il est configuré pour la communication RS232.
  - Vérifiez sur le KC601 que le LED vert assigné au port de configuration RS232 (se référer à la *Figure 1*, le LED vert dans la section supérieure de gauche) n'est pas constamment allumé (sans clignotement). Si c'est le cas, redémarrez le KC601 en appuyant sur le bouton poussoir pour le « Reset » (se référer à la *Figure 1*, à l'extrémité gauche, vers le haut) avant d'effectuer une nouvelle tentative de lecture de la configuration.

- Si vous utilisez un convertisseur USB/RS232, alors il y a de fortes chances que Windows ait choisi automatiquement le mauvais pilote (*« driver »*) pour gérer le convertisseur. Remplacer ce mauvais pilote par celui fourni avec le convertisseur risque de ne pas être suffisant; il faut désinstaller le mauvais pilote avec un logiciel comme : *« DRemover98_2K.exe »* fourni sur le CD accompagnant le convertisseur : *« USB to Serial Port Cable»* de marque « ***nexXtech* »,** modèle ***« 2608042 ».***

Transmettre Config. au KC601

- o Ce bouton permet d'enclencher le transfert de la configuration à l'écran vers le KC601, en autant que le champ du *« baud rate »* ait une valeur non nulle.
- o À partir du moment où ce bouton est pressé, les champs et les boutons de la fenêtre de configuration demeurent bloqués tant que la requête n'est pas complétée, avec ou sans succès.
  - Si le KC601 reçoit une configuration valide avec au moins un nouveau paramètre de configuration, alors la nouvelle configuration reçue sera écrite dans la mémoire flash (le LED jaune se mettra alors à clignoter très rapidement tant que l'écriture en flash ne sera pas complétée), puis le KC601 se réinitialisera pour appliquer la nouvelle configuration; se référer à la section : *2.1.3 LED / Diodes Électroluminescentes*.
  - Si le KC601 reçoit une configuration valide avec les mêmes paramètres déjà présents dans le KC601, alors il n'y aura ni écriture dans la mémoire flash, ni réinitialisation.
  - Si la nouvelle configuration n'était pas valide ou si une erreur est survenue au cours du transfert, alors la nouvelle configuration (si elle s'est rendue au KC601) sera rejetée et un message d'erreur vous avisera de la situation.
  - Si vous n'avez pas de réponse du KC601, un message vous en informera au bout de quelques secondes.
- o Si vous éprouvez des problèmes de communication avec le KC601, se référer aux solutions proposées pour régler ce problème lors de la réception de la configuration du KC601.

#### 3.1.4.3 Échange de configuration avec une unité de sauvegarde (habituellement le disque dur)

Enregistrer Config. sur Disque

- o Vous pouvez enregistrer dans un fichier les paramètres de configuration affichés. Seul le format de version 1 est permis actuellement, les versions antérieures n'étant plus supportées puisqu'elles doivent être remplacées par la nouvelle version.

Lire Config. du Disque Dur

- o Vous pouvez lire sur disque les paramètres de configuration enregistrés dans un fichier via le bouton précédent.
  - Si la configuration retrouvée est valide, seule la version 1 étant acceptée actuellement, elle est affichée à l'écran.
  - Dans le cas contraire, un message d'erreur vous indiquera pourquoi la configuration a été rejetée.

#### 3.1.4.4 Réinitialisation des paramètres à l'écran




- o Ce bouton remettra les paramètres à l'écran aux valeurs par défaut affichées à la *Figure 4*.

## 3.2 Onglet « *Communication* » / Fenêtre des statistiques sur la communication perçue par le port *BACnet* du KC601

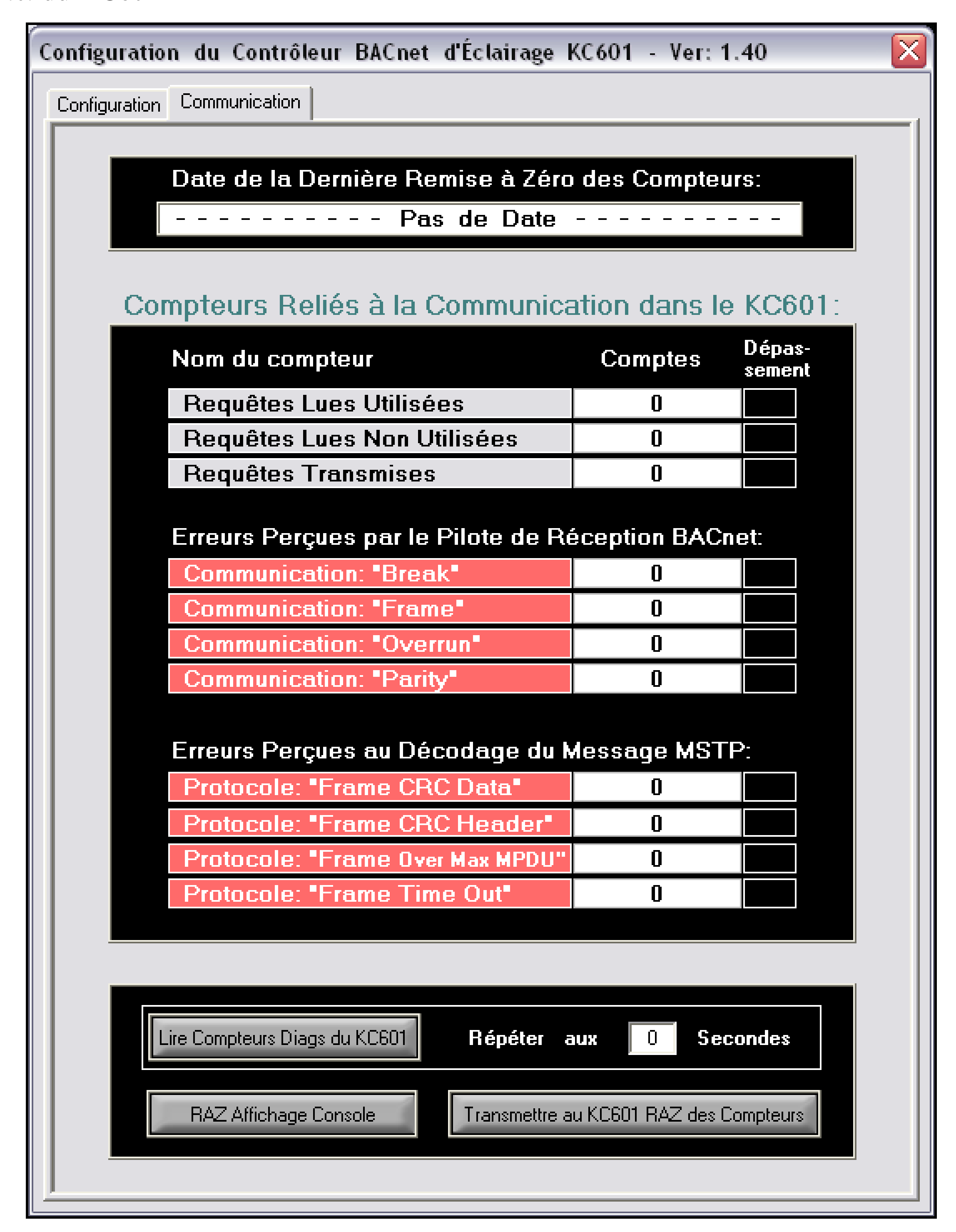



*Figure 10 : Fenêtre sur la communication BACnet lors de sa première ouverture.*

Le contrôleur d'éclairage possède plusieurs compteurs pour maintenir des statistiques sur la qualité de la communication *BACnet* telle que perçue par le port RS485 affecté à cet effet sur le KC601 (se référer à la section de droite de la *Figure 1*). La partie du bas de cette fenêtre de la console de configuration permet d'extraire les statistiques du KC601 et de les afficher à l'écran.

### 3.2.1 Bloc d'entête de la fenêtre sur la communication *BACnet*




*Figure 11 : Section du haut de la fenêtre sur la communication BACnet perçue par le KC601.*

Cette section affiche la date de la dernière requête transmise à partir de cette fenêtre de communication via le bouton : *« Transmettre au KC601 RAZ des Compteurs »*.

- L'affichage sans date de la *Figure 11* est obtenu :
  - au démarrage de la console de configuration lorsqu'aucune requête de lecture n'a encore été faite via le bouton : *« Lire Compteurs Diags du KC601 »*;
  - après une requête de lecture des compteurs du KC601 mais :
    - le KC601 venait d'être redémarré ou d'être réinitialisé à partir du bouton poussoir pour le *« Reset »* (coin supérieur gauche de la *Figure 1*) ou par l'intermédiaire d'une requête *BACnet : « Cold Start »*;
    - le KC601 n'a pas encore reçu de requête via le bouton : *« Transmettre au KC601 RAZ des Compteurs »* depuis la dernière réinitialisation du contrôleur.

- Lorsque vous transmettez une requête via le bouton : *« Transmettre au KC601 RAZ des Compteurs »*, la date et l'heure de votre ordinateur au moment de la transmission est enregistrée dans le KC601 avant qu'il ne réinitialise ses compteurs statistiques sur la communication *BACnet*. Une requête de lecture via le bouton *« Lire Compteurs Diags du KC601 »* fournira la date de la dernière requête RAZ (de *« Remise À Zéro »*) :

Date de la Dernière Remise à Zéro des Compteurs:
Jeudi le 01 Octobre 2009 - Temps = 17:59:26

*Figure 12 : Exemple du format de date retourné par le KC601 pour de la dernière remise à zéro des compteurs.*

### 3.2.2 Bloc d'entête de la fenêtre sur la communication *BACnet*

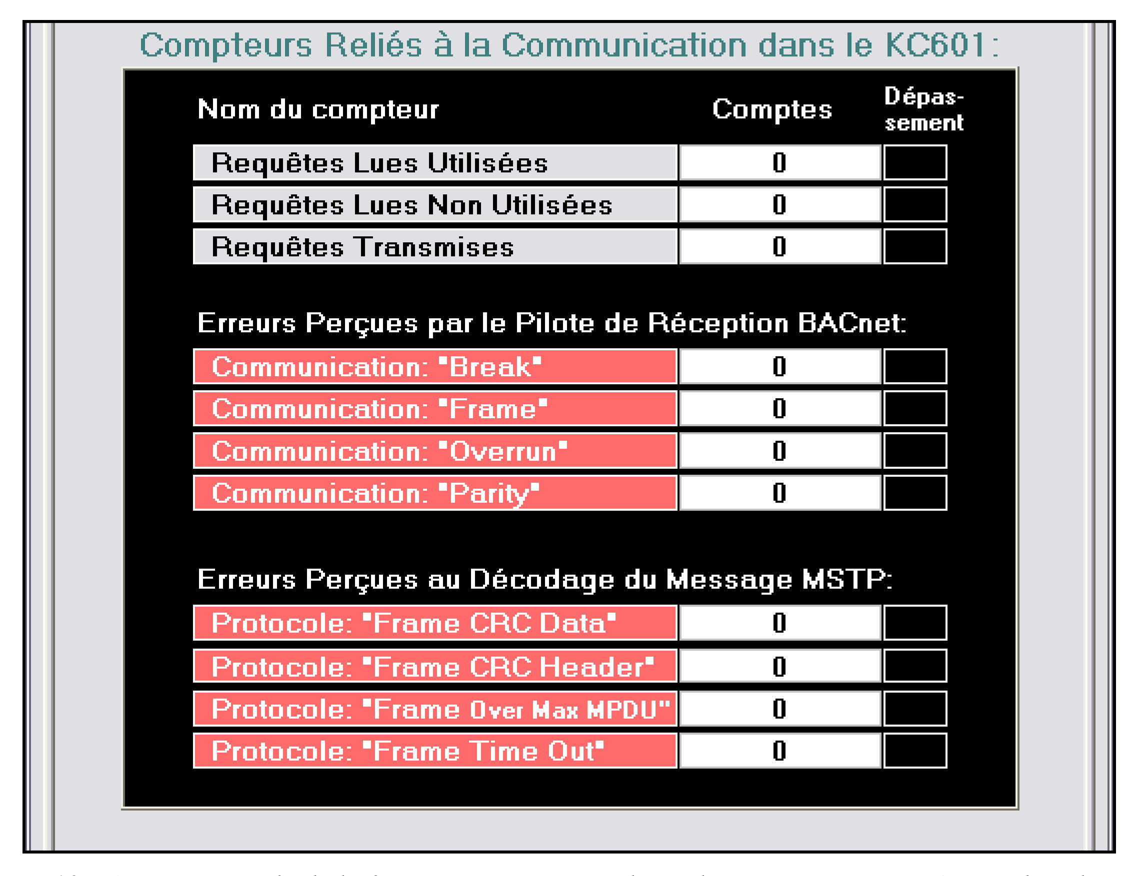



*Figure 13 : Section centrale de la fenêtre : compteurs reliés à la communication BACnet dans le KC601.*

Tous les compteurs qui surveillent la qualité de la communication *BACnet* dans le KC601 apparaissent dans cette section.

- Les compteurs avec une étiquette rouge signalent les erreurs rencontrées.
- La colonne « *Comptes* » indique le nombre de requêtes détectées par le port RS485 *BACnet* du KC601 correspondant au critère indiqué par l'étiquette à la gauche de compteur.
- Sous la colonne avec le titre « *Dépassement* », la case noire sera remplacée par : Over si le compteur est revenu à zéro (« *wrap around* ») pour poursuivre son incrémentation après avoir atteint la valeur la plus élevée permise pour son compte (0xFFFFFFFF ➔ 4294967295).

#### 3.2.2.1 Compteurs de transactions *BACnet - MS/TP*

| Nom du compteur | Comptes | Dépas-sement |
|---|---|---|
| Requêtes Lues Utilisées | 262828 | |
| Requêtes Lues Non Utilisées | 456481 | |
| Requêtes Transmises | 390641 | |

*Figure 14 : Fenêtre sur la communication BACnet – Compteurs de requêtes BACnet.*

On parle de communication *BACnet* mais pour plus de précision, les compteurs de cette section s'incrémentent uniquement en fonction de la couche *MS/TP* (voir la section *1.1 Acronymes et abréviations*) qui gère les échanges entre les diverses composantes du réseau *BACnet.* La distinction vient du fait que la couche *MS/TP* peut contenir ou non une requête *BACnet* mais une requête *BACnet* passe assurément par la couche *MS/TP*.

Pour les 2 compteurs de *« requêtes lues »*, ils sont incrémentés si aucune erreur n'est survenue au niveau de la couche *MS/TP*. Le total des requêtes *MS/TP* acceptées sur le port RS485 *BACnet* s'obtient en additionnant les valeurs affichées par ces 2 premiers compteurs.

- ***« Requêtes Lues Utilisées »*** **:** requêtes *MS/TP* qui s'adressaient au KC601, soit parce que l'adresse réseau du KC601 était visée, soit parce qu'il s'agissait d'une requête *« broadcast »,* s'adressant à tous les *« BACnet devices »* sur la même branche du réseau que le KC601.
- ***« Requêtes Lues Non Utilisées »*** **:** requêtes *MS/TP* qui ne s'adressaient pas au KC601.
- ***« Requêtes Transmises »*** **:** requêtes *MS/TP* émises par le KC601, que ce soit :
  - en réponse à une requête *BACnet* reçue (pour transmettre la composition des objets *BACnet* ou leurs propriétés),
  - pour transmettre une requête *BACnet « I_AM »* au démarrage, de façon cyclique ou en réponse à un message *« WHO_IS »* pour une étendue d'identificateurs *BACnet* à laquelle correspond le KC601,
  - pour échanger le jeton,
  - pour découvrir d'autres nœuds…

#### 3.2.2.2 Compteurs d'erreurs perçues par le pilote du port RS485 affecté à la communication *BACnet*

| Erreurs Perçues par le Pilote de Réception BACnet: | | |
|---|---|---|
| Communication: "Break" | 0 | |
| Communication: "Frame" | 0 | |
| Communication: "Overrun" | 0 | |
| Communication: "Parity" | 0 | |

*Figure 15 : Fenêtre sur la communication BACnet – Compteurs d'erreurs du pilote du port RS485.*

Ces compteurs sont incrémentés chaque fois qu'une erreur est détectée au niveau de la réception du port de communication *RS485 BACnet* sur le KC601, lorsque le logiciel « pilote » (le *« driver »* du port *RS485 BACnet)* décortique les bits et les octets un à un, avant même qu'ils n'aient été validés pour former un message *MS/TP*.

- ***« Communication: "Break" » :*** cette erreur est détectée si le *« start bit »* dure plus longtemps que la largeur (temporelle) du « cadre » (*« frame »*) permis pour le *« baud rate »* courant, donc dure plus longtemps que le temps alloué à la réception de tous les bits d'un octet.
- ***« Communication: "Frame" » :*** lorsque l'octet reçu n'entre pas dans le « cadre » prévu pour le baud rate choisi, i.e. en incluant tous les bits qui composent l'octet reçu, le *« start bit »* et le *« stop bit »* qui encadrent l'octet n'entrent pas dans la fenêtre de temps permise pour la vitesse de communication (*« baud rate »*) courante.
- ***« Communication: "Overrun" » :*** erreur d'écrasement : un caractère en attente n'a pu être lu avant l'arrivée du caractère suivant. Cette erreur indique donc qu'au moins un octet a été perdu.
- ***« Communication: "Parity" » :*** ce mécanisme de validation requiert l'ajout d'un *« bit de parité »* pour chaque octet transmit, ce bit servant à obtenir un nombre pair ou impair de bits, selon la validation de parité choisie (s'il y a lieu). Une erreur de parité indique la corruption de l'octet reçu. Les ports de communication doivent avoir été configurés de la même manière aux 2 extrémités pour que le processus de validation fonctionne…

#### 3.2.2.3 Compteurs d'erreurs perçues au moment du décodage du message *MSTP*

| Erreurs Perçues au Décodage du Message MSTP: | | |
|---|---|---|
| Protocole: "Frame CRC Data" | 0 | |
| Protocole: "Frame CRC Header" | 0 | |
| Protocole: "Frame Over Max MPDU" | 0 | |
| Protocole: "Frame Time Out" | 0 | |

*Figure 16 : Fenêtre sur la communication BACnet – Compteurs d'erreurs du protocole MS/TP.*

Ces compteurs sont incrémentés chaque fois qu'une erreur est détectée au niveau de la validation de chacun des octets reçus (déjà validés au niveau du *« driver »* RS485) pour former un message *MS/TP*. Dans la fenêtre des compteurs, ils sont affichés selon un ordre alphabétique et non selon l'ordre dans lequel les erreurs peuvent survenir.

- ***« Protocole: "Frame CRC Header" » :*** cette erreur est signalée si l'addition de tous les octets reçus dans l'entête du message qui précède les octets de données ne correspond pas à une valeur pré-établie (0x55).
- ***« Protocole: "Frame CRC Data" » :*** cette erreur est signalée seulement si le message contient des données (i.e. un message *BACnet*) et que l'addition de tous les octets de données reçus (à la suite de l'entête du message), en incluant les 2 octets complémentaires (*CRC1* et *CRC2*) aux données, ne correspond pas à une valeur pré-établie (0xF0B8).
- ***« Protocole: "Frame Over Max MPDU" » :*** cette erreur est signalée si, après avoir reçu l'entête du message, trop d'octets de données suivent, i.e. le nombre d'octets de données reçus dépasse la capacité du *« buffer »* de traitement du KC601 (ce qui risque de ne jamais arriver).
- ***« Protocole: "Frame Time Out" » :*** cette erreur est signalée autant lors de la réception des octets qui forment l'entête du message que pour les octets de données qui peuvent compléter le message. Elle survient si un octet prend trop de temps à entrer, si le « compteur de temps associé à la durée du silence » dépasse la valeur permise entre 2 octets.

### 3.2.3 Section des boutons pour initier les requêtes




*Figure 17 : Fenêtre sur la communication BACnet – Compteurs d'erreurs du protocole MS/TP.*

Comme pour la fenêtre principale pour configurer le KC601 (*Figure 4*), les boutons pour déclancher les processus se retrouvent dans le bas de l'écran. Si la console est incapable de communiquer avec le KC601, vous en serez avisés par un message :




*Figure 18 : Fenêtre sur la communication BACnet – Message pour signaler un problème de communication.*

Se référer aux solutions de dépannage proposées à la section *3.1.4.2 Échange de configuration avec le KC601* si vous éprouvez des problèmes de communication avec le port RS232.




- Bouton pour initier une requête de lecture des paramètres de diagnostique via le port RS232, afin d'afficher l'état actuel des compteurs statistiques du KC601, soit la section affichée par la *Figure 13*.
- 

  - Si vous avez laissé ce paramètre à zéro, lorsque l'on appuie sur le bouton *« Lire Compteurs Diags du KC601 »,* une requête simple sera transmise pour recevoir et faire afficher la valeur courante des compteurs dans le KC601.
  - Pour une valeur *« X »* non nulle : lorsque l'on appuie sur le bouton *« Lire Diags »*, le texte de ce bouton change pour : ANNULER Lecture Diags .

    Tant que vous ne cliquerez pas sur ce bouton pour annuler les requêtes, il y aura des transmissions aux *« X »* secondes vers le KC601 pour faire rafraîchir la fenêtre avec les valeurs courantes des compteurs.

Les requêtes cesseront d'elles-mêmes cependant dans le cas où la console serait incapable de communiquer avec le KC601, avec un message pour vous aviser du problème (voir la *Figure 18*).

Transmettre au KC601 RAZ des Compteurs

- Ce bouton transmet au KC601 une requête « RAZ » pour **R**emettre **À** **Z**éro tous les compteurs « statistiques » **dans le KC601**. La requête fournit aussi au KC601 la date et l'heure courantes de l'ordinateur pour que le KC601 l'enregistre dans sa mémoire volatile.
- Cette requête de remise à zéro ne se reflète pas immédiatement à l'écran de la console. Une requête de lecture via le bouton *« Lire Compteurs Diags du KC601 »* est nécessaire pour rafraîchir l'état des compteurs et de la date résultant de la transmission du *« Reset »* des compteurs dans le KC601.




- Ce bouton ne transmet rien au KC601 de sorte qu'il n'influence pas le contenu des compteurs. Il ne fait que réinitialiser l'affichage de la fenêtre sous l'onglet *« Communication »*, ce qui résulte en une remise à zéro de toutes les valeurs à l'**écran** (exception faite de la période pour répéter les requêtes de lecture des compteurs).

# 4. Contrôleurs de relais KC401 / Programmation des groupes

## 4.1 Résumé du fonctionnement du contrôleur de relais KC401

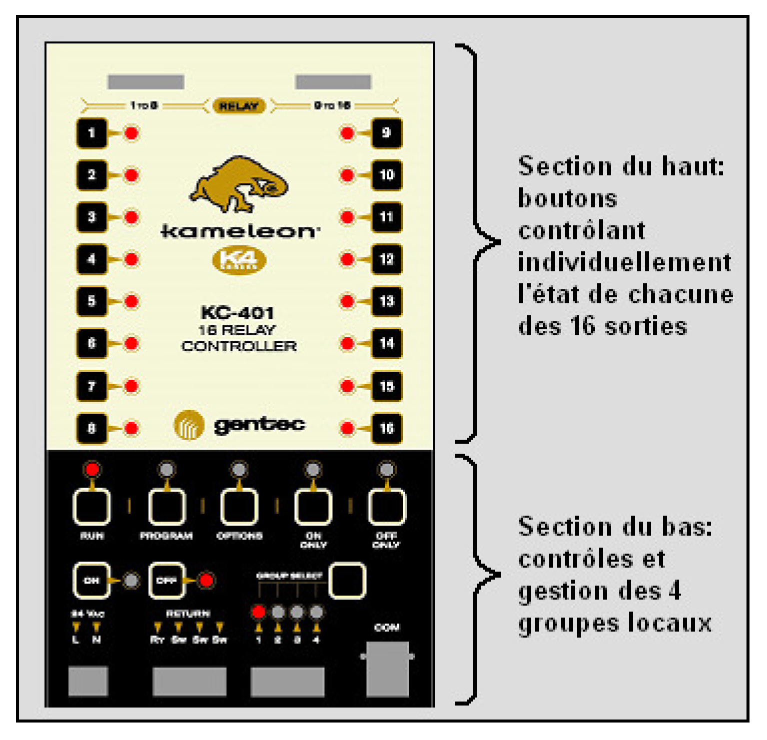



*Figure 19 : Sections d'un contrôleur KC401 à brancher au contrôleur BACnet KC601*

**NOTE :** Pour une version plus détaillée sur le fonctionnement du KC401, lire le document portant spécifiquement sur le KC401 : *« 121-35701 - Manuel d'installation, d'opération et d'entretien du contrôleur d'éclairage KC401 ».*

- Chaque KC401 est composé de :
  - 16 sorties/relais qui peuvent être activés individuellement sur le KC401 à partir des boutons de la subdivision du haut (au-dessus de la section noire du KC401; se référer à la *Figure 19*).
  - 4 groupes par KC401; avec le contrôleur *BACnet* KC601, chaque groupe peut être composé :
    - … d'un maximum de 64 sorties lorsque quatre KC401 sont reliés au KC601;
    - … d'options *(« TOE », « Warning », « ON Only », « OFF Only »)* applicables aux sorties du groupe lorsqu'il y a une requête de mise en marche *(« ON »)* ou d'arrêt *(« OFF »)* des sorties du groupe (voir la section : *4.4 Options pour les sorties des KC401*).

    Le choix et l'activation des groupes se fait à partir de la section (noire) du bas du KC401, en se référant à la *Figure 19*.

- Un KC401 a deux modes de fonctionnement de base :

  - ***Mode « PROGRAM » :***

    Ce mode permet de programmer les groupes de sorties; se référer à la section : *4.2 Programmation de groupes*.

    - Si le KC401 était en mode *« RUN »* (le LED correspondant étant allumé au-dessus du bouton de même nom), appuyer sur le bouton *« PROGRAM »* fait passer tous les KC401 et le contrôleur *BACnet* KC601 en mode *« PROGRAM »*.
      - Le LED au-dessus du bouton *« PROGRAM »* reste allumé pour le KC401 d'où provient la requête, ce dernier étant alors appelé ***« KC401 maître »***. Le choix du groupe à créer et ses options se feront à partir de ce KC401.
      - Ce même LED clignote pour tous les autres KC401 reliés via le KC601, ces derniers étant asservis au *« KC401 maître »* portent alors l'appellation ***« KC401 esclaves »*** puisqu'ils se limitent à refléter lesquelles parmi leurs sorties appartiennent au groupe couramment choisi sur le *« KC401 maître »*.
    - Si le KC401 était déjà en mode *« PROGRAM »* et que l'on appuie à nouveau sur le bouton *« PROGRAM »* :
      - s'il s'agit du *« KC401 maître »*, alors 2 LED clignoteront un bref moment parmi ceux des sorties individuelles de ce KC401 (section du haut de la *Figure 19*), les 2 chiffres correspondants indiquant quelle est la version du logiciel qui contrôle le KC401;
      - s'il s'agit d'un *« KC401 esclave »*, alors cela n'aura aucun effet.

    Un compteur limite la durée du mode *« PROGRAM »* : si 2 minutes s'écoulent sans qu'aucune touche ne soit pressée sur l'un ou l'autre des KC401 alors qu'ils sont en mode *« PROGRAM »*, le KC601 et tous les KC401 retourneront automatiquement au mode *« RUN »*.

  - ***Mode « RUN » :***

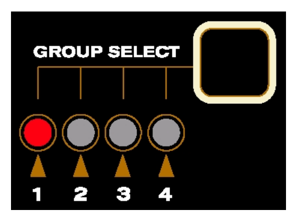
    Ce mode permet d'activer ou d'éteindre des groupes de sorties ou des sorties individuelles; se référer à la section :

    *4.3 Activation de groupes et de sorties* individuelles. Appuyer sur le bouton *« RUN »* n'aura d'effet que si le KC401 est en mode *« PROGRAM »* et qu'il s'agit du *« KC401 maître »*. Ces 2 conditions étant respectées, tous les KC401 (et le contrôleur KC601) retourneront alors en mode *« RUN »*.

## 4.2 Programmation de groupes

Un groupe est un agencement de sorties individuelles choisies parmi tous les KC401 reliés à un même contrôleur *BACnet* d'éclairage KC601. Un groupe permet l'envoi d'une commande unique pour allumer ou éteindre un ensemble de relais.

**Choix du groupe à partir d'un KC401 :**




Que ce soit en mode *« PROGRAM »* ou en mode *« RUN »*, le choix d'un groupe sur un KC401 se fait avec le bouton *« GROUP SELECT »* dans la partie noire du KC401 (en se référant à la *Figure 19*). En mode *« PROGRAM »* cependant, ce choix ne peut se faire qu'à partie du *« KC401 maître »* (voir *« Mode Program »* dans : *4.1 Résumé du fonctionnement du contrôleur de relais* ).

Tel qu'énoncé à la section : *2.1.2 Ports de communication*, le contrôleur *BACnet* identifie les KC401 selon l'entrée du contrôleur *BACnet* à laquelle le KC401 est relié. Ainsi :

- le KC401 branché dans l'entrée 1 du contrôleur permet de choisir/programmer les groupes 1, 2, 3 et 4, de même que le choix des sorties individuelles de 1 à 16 inclusivement;
- le KC401 branché dans l'entrée 2 du contrôleur permet de choisir/programmer les groupes 5, 6, 7 et 8, de même que le choix des sorties individuelles de 17 à 32 inclusivement, etc.

**Programmation du groupe à partir d'un KC401 :**

On ne peut programmer un groupe qu'à partir des KC401.

- Tel qu'énoncé en *4.1 Résumé du fonctionnement du contrôleur de relais* , le passage d'un KC401 du mode *« RUN »* au mode *« PROGRAM »* fait passer les autres KC401 en mode *« PROGRAM »,* le premier devenant le *« maître »*, les autres les *« esclaves »* du premier.
- Sur chacun des KC401, les sorties composant le groupe couramment choisi sont indiquées (dans la sub-division du haut) par un LED allumé vis-à-vis les boutons avec un numéro de sortie (entre 1 et 16).
- Sur le *« KC401 maître »*, tous les boutons sont disponibles :
  - *« RUN »* : pour faire sortir tous les KC401 (et le KC601) du mode *« PROGRAM »*.
  - *« OPTIONS » :* pour choisir une combinaison des options *« TOE »* et *« WARNING »* pour le groupe couramment choisi; se référer à la section : *4.4 Options pour les sorties des KC401.*
  - *« ON ONLY » :* option que l'on peut associer au groupe pour qu'il ne réponde qu'aux requêtes de mise à *« ON »* du groupe. Le choix de cette option désactive l'option *« OFF ONLY »* si elle a été choisie. Se référer à la section : *4.4 Options pour les sorties des KC401.*
  - *« OFF ONLY » :* option que l'on peut associer au groupe pour qu'il ne réponde qu'aux requêtes de mise à *« OFF »* du groupe. Le choix de cette option désactive l'option *« ON ONLY »* si elle a été choisie. Se référer à la section : *4.4 Options pour les sorties des KC401.*
  - *« ON » :* en mode *« PROGRAM »*, les 16 sorties du KC401 sont toutes choisies pour faire partie du groupe courant.
  - *« OFF » :* en mode *« PROGRAM »*, les 16 sorties du KC401 sont toutes éliminées du groupe courant.

- *« GROUP SELECT »* : pour choisir un autre groupe à programmer parmi les 4 groupes du *« KC401 maître »*. Lorsque l'on change de groupes, les LED des sorties individuelles (au-dessus de la section noire de la *Figure 19*) sur tous les KC401 reflètent la composition du nouveau groupe choisi.
- Boutons des sorties individuelles (numérotés de 1 à 16 dans la subdivision du haut) pour ajouter (LED *« ON »*) ou enlever (LED *« OFF »*) une sortie du groupe courant.

o Sur les *« KC401 esclaves »*, l'accès au clavier se limite à la sélection des sorties devant faire partie du groupe couramment choisi sur le *« KC401 maître »*. Ainsi, seuls sont disponibles les boutons groupes *« ON »* et *« OFF »* (dans la section noire du bas pour les groupes sur la *Figure 19*) et ceux pour les sorties individuelles (au-dessus de la subdivision noire de la *Figure 19*)

o La composition de chacun des groupes, de même que leurs options, ne sont enregistrées au complet que dans la passerelle KC601.

## 4.3 Activation de groupes et de sorties individuelles

Les KC401 doivent être en mode *« RUN »* pour que l'on puisse changer l'état des relais branchés aux KC401. Se référer à la section : *4.4 Options pour les sorties des KC401* pour savoir quand certaines requêtes seront barrées ou appliquées différemment selon les options en cours.

o L'activation (*« ON »* ou *« OFF »*) des groupes peut être initiée à 3 niveaux :
  - À partir des boutons *« ON »* ou *« OFF »* au bas d'un KC401 (section noire de la *Figure 19*), le groupe visé étant celui couramment choisi sur ce KC401 *« maître »*, dans le bloc *« GROUP SELECT »*. Pour ces requêtes, les options *« TOE »* et *« WARNING »* ne s'appliquent pas.
  - À partir d'une entrée d'un KC401, habituellement déclanchée par un interrupteur, chacune des 4 entrées étant associée à l'un des 4 groupes du KC401. Pour ces requêtes, toutes les options s'appliquent.
  - Requête *BACnet* issue d'une console *BACnet* pour mettre une *« sortie binaire »*, associée à un groupe, à *« ON » / « ACTIVE »* ou à *« OFF » / « INACTIVE »*; se référer à la section : *5.1.2 « Binary Output Object Type »*. Pour ces requêtes, toutes les options s'appliquent.

o L'activation des sorties individuelles peut être initiée à 3 niveaux :
  - À partir des boutons dans la subdivision du haut du KC401 (au-dessus de la section noire de la *Figure 19*), numérotés de 1 à 16, pour allumer (LED *« ON »*) ou éteindre (LED *« OFF »*) un relais.
  - À partir d'un interrupteur qui contrôle un relais connecté à une sortie individuelle du KC401.
  - Requête *BACnet* issue d'une console *BACnet* pour mettre une *« sortie binaire »*, associée à un sortie individuelle, à *« ON » / « ACTIVE »* ou à *« OFF » / « INACTIVE »*; se référer à la section : *5.1.2 « Binary Output Object Type »*.

## 4.4 Options pour les sorties des KC401

Dans sa version actuelle, le contrôleur *BACnet* d'éclairage KC601 ne supporte pas les options pour les groupes. La description des options sera donc ajoutée avec la prochaine version du KC601…

# 5. La carte de contrôle KC601 en tant qu'objet *BACnet*

## 5.1 Types d'objets *BACnet* et propriétés supportées

Dans cette section, les propriétés présentées pour chacun des objets *BACnet* apparaissent selon l'ordre dans lequel elles sont énumérées dans le manuel de la norme *BACnet*. Elles sont toutes accessibles en lecture par une requête *BACnet* pour lire une seule ou plusieurs propriétés à la fois. À la droite de chacune des propriétés, on peut retrouver entre parenthèses les lettres suivantes :

**O :** propriété **O**ptionnelle (facultative), non exigée par la norme *BACnet*.

**R :** propriété **R**equise (obligatoire), exigée par la norme *BACnet*.

**W :** propriété modifiable en écriture *(« **W**RITE »)*, à partir d'une console *BACnet*.

**NOTE :** les paramètres qui peuvent être modifiés via une console *BACnet* seront perdus après une réinitialisation (requête *« cold restart »*, bouton *« reset »* sur la carte KC601 ou en alimentant la carte après l'avoir éteinte) et seront alors remplacés par les valeurs par défaut.

### 5.1.1 « Device Object Type »

Le KC601 est détecté dans une console *BACnet* en tant que *« device object »*. Certains paramètres qui caractérisent le KC601 ne font partie d'aucune propriété *BACnet*, peu importe le type d'objet :

- l'*adresse réseau (« Network Device ID »)*,
- le *mot de* passe,
- le *« baud rate »* pour la communication *BACnet.*

Ainsi, bien que les 2 premiers soient affichés dans une console *BACnet* avec l'objet *« device »*, ils ne sont pas énumérés par la norme *BACnet* pour caractériser l'objet de type *« device »*. Ces 3 paramètres peuvent être établis à l'aide de la console de configuration *Windows*; se référer à la section : *3.1.2.1 Paramètres de base pour établir le KC601 en tant qu'objet BACnet.*

**Propriétés supportées :**

- *« Object_Identifier »* (R) : le numéro d'identification de la passerelle en tant qu'objet *BACnet*. Bien que ce paramètre ne puisse être modifié par une requête d'écriture *BACnet*, on peut le changer à l'aide de la console de configuration *Windows*; se référer à la section : *3.1.2.1 Paramètres de base pour établir le KC601 en tant qu'objet BACnet*.
- *« Object_Name »* (R) : le nom (caractères alphanumériques) attribué à la passerelle = *"Ctrl. Eclairage"* ou *"Lighting Ctrl."*, selon la langue choisie pour les descriptions à partir du configurateur.
- *« Object_Type »* (R) : le type d'objet BACnet = *"DEVICE"*.
- *« System_Status »* (R) : l'état courant de fonctionnement de la passerelle. La seule valeur retournée est : *"STATUS_OPERATIONAL"*.
- *« Vendor_Name »* (R) : le nom du manufacturier de la passerelle KC601 = *"GENTEC"*.
- *« Vendor_Identifier »* (R) : le code d'identification unique assigné par *ASHRAE* pour identifier le manufacturier *« GENTEC » = "285"*.
- *« Model_Name »* (R) : le numéro de modèle de notre appareil *BACnet = "KC601"*.
- *« Firmware_Revision »* (R) : la valeur retournée est la version du *« stack BACnet »* intégré avec le code de la passerelle KC601 pour la communication *MS/TP*.
- *« Application_Software_Version »* (R) : la version courante de l'application logicielle qui fait fonctionner le KC601.

- *« Location »* (O/W) : texte pour indiquer l'endroit où est installée le KC601. On peut utiliser jusqu'à 474 caractères (en incluant le caractère de fin de chaîne) = *(MAX_APDU - 6)*. Par défaut, la chaîne est initialisée au lieu où est situé le manufacturier = *"Quebec Canada"*.
- *« Description »* (R/W) : texte pour décrire le KC601 ou l'utilisation qui en est faite. On peut utiliser jusqu'à 474 caractères (en incluant le caractère de fin de chaîne) = *(MAX_APDU - 6)*. Par défaut, la chaîne décrit en quoi consiste le KC601 = *"Controleur d'Eclairage BACnet pour 4 KC401"* ou *"BACnet Lighting Controller for 4 KC401",* selon la langue choisie pour les descriptions à partir du configurateur.
- *« Protocol_Version »* (R) : indique le numéro de révision « majeur » de la dernière version du protocole *BACnet* supportée par le KC601.
- *« Protocol_Revision »* (R) : indique le numéro de révision « mineur » de la dernière version du protocole *BACnet* supportée par le KC601
- *« Protocol_Services_Supported »* (R) : liste des services standards *BACnet* fournis par le KC601; se référer à la section : *5.2 Commandes/Services BACnet supportés (selon* les *modèle BIBBs / BACnet Interoperability Building Blocks – Annexe K de la norme ASHRAE)*
- *« Protocol_Object_Types_Supported »* (R) : liste des divers types d'objets *BACnet* supportés par le KC601, soit ceux énumérés dans cette section 5.1.
- *« Object_List »* (R) : cette propriété fournit la liste de tous les objets supportés dans le KC601, en incluant le type et le numéro d'identification *(« Object_Identifier »)* de chacun des objets.
- *« Max_APDU_Length_Accepted »* (R) : nombre maximum d'octets pouvant être contenus dans une unité de données, indivisible, de la couche d'application du protocole *(Application layer Protocol Data Unit).* Dans notre système, cette valeur a été fixée à *"480"*.
- *« Segmentation_Supported »* (R) : indique si la segmentation de message est supportée et si c'est le cas, est-elle supportée en transmission, en réception ou les deux en même temps. Dans le cas qui nous concerne, comme le KC601 ne supporte pas cette option, la valeur retournée = *"NO_SEGMENTATION"*.
- *« APDU_Timeout »* (R/W) : temps requis en millisecondes avant de retransmettre un *APDU* requérant une réponse, lorsque cette réponse n'est pas arrivée. La valeur établie par défaut = *"3000"*.
- *« Number_Of_APDU_Retries »* (R/W) : indique le nombre maximum de fois que l'on doit retransmettre l'*APDU* (lorsque la réponse attendue n'arrive pas). La valeur établie par défaut = *"3"*.
- *« Max_Master »* (R/W) : cette propriété est requise dans notre contexte parce que le KC601 agit en *« nœud maître »* dans un réseau de communication *MS/TP*. Cette propriété établit l'adresse réseau la plus élevé pour communiquer avec d'autres nœuds maîtres. Cette valeur peut être modifiée via le configurateur du KC601 (section *3.1.2.3 Paramètres ajustables du protocole BACnet*).
- *« Max_Info_Frames »* (R/W) : cette propriété est requise dans notre contexte parce que le KC601 agit en *« nœud maître »* dans un réseau de communication *MS/TP*. Cette propriété permet de spécifier le nombre maximum de *« trames d'information »* que le nœud peut transmettre avant de devoir passer le *« jeton » (« token »).* Cette valeur peut être modifiée via le configurateur du KC601 (section *3.1.2.3 Paramètres ajustables du protocole BACnet*).
- *« Device_Address_Binding »* (R) : il s'agit d'une liste d'identificateurs *BACnet (« Object_Identifier »)* pour les *« BACnet Devices »* reliés à des « adresses *BACnet* » sur le réseau par lesquels on peut communiquer avec ces objets. La liste retournée dans notre cas est vide car sur le KC601, les autres types d'objets *BACnet* que l'on y retrouve n'ont pas d'adresse réseau et le KC601 agit en *« nœud maître »* principalement pour pouvoir répondre aux requêtes *« WhoIs »* pour la détection automatique des appareils *BACnet* sur le réseau.

- *« Database_Revision »* (R) : ce nombre est incrémenté chaque fois qu'un objet est créé, détruit, qu'un nom d'objet est modifié, qu'un identificateur *BACnet (« Object_Identifier »)* est changé ou que les valeurs par défaut sont rétablies.

  Pour le KC601, l'incrémentation aura lieu lorsque certains changements seront apportés via la console de configuration : changement du nombre de KC401, cocher ou décocher une option d'affichage d'objets *BACnet*, nouvel identificateurs *BACnet.*

---

### 5.1.2 « Binary Output Object Type »

C'est sous ce type d'objet que les groupes de sorties et les sorties individuelles des KC401 reliés au KC601 sont affichés dans une console *BACnet*, en autant que le contrôleur KC601 ait été configuré pour retourner ces informations; se référer à la section : *3.1.2.2 Sélections des objets BACnet du KC601 accessibles via une console BACnet.*

**Propriétés supportées :**

- *« Object_Identifier »* (R) : le numéro d'identification de chacun de ces objets *BACnet* intégrés au KC601. Ce numéro est fixe et débute à 1 pour être incrémenté avec chaque *« binary output »*. Ainsi, chaque groupe a un *« Objext Identifier »* qui correspond au numéro du groupe. Si des sorties individuelles sont affichées, le numéro suivant celui du dernier groupe lui sera attribué et ainsi de suite.
- *« Object_Name »* (R) :
  - le KC601 attribue aux groupes des noms (caractères alphanumériques) immuables selon le modèle : *« BIN_GROUP_XX »* où *XX* est le numéro du groupe (valeur entre 01 et 16) dans la passerelle, débutant par un zéro pour les chiffres inférieurs à 10;
  - le même principe s'applique pour les noms de sorties individuelles, selon le modèle : *" BIN_OUTPUT_XX"*, où *XX* est le numéro de la sortie (valeur entre 01 et 64) dans la passerelle.

| | Output Name | Description |
|---|---|---|
| 1 ▶ | BIN_GROUP_01 | Groupe No.1 du K4 No.1 |
| 2 | BIN_GROUP_02 | Groupe No.2 du K4 No.1 |
| 3 | BIN_GROUP_03 | Groupe No.3 du K4 No.1 |
| 4 | BIN_GROUP_04 | Groupe No.4 du K4 No.1 |
| 5 | BIN_OUTPUT_01 | Sortie No.01 du K4 No.1 |
| 6 | BIN_OUTPUT_02 | Sortie No.02 du K4 No.1 |
| 7 | BIN_OUTPUT_03 | Sortie No.03 du K4 No.1 |
| 8 | BIN_OUTPUT_04 | Sortie No.04 du K4 No.1 |

*Figure 20 : Exemple d'affichage des noms générés par le KC601 (en français) pour ses objets de type « Binary Output »*

- *« Object_Type »* (R) : le type d'objet BACnet = *"BINARY_OUTPUT"*.
- *« Present_Value »* (R/W) : l'état dans lequel se trouve l'objet *BACnet*. Si le groupe ou la sortie individuelle est allumée, la valeur de cette propriété est : *« ACTIVE »* et *« INACTIVE »* lorsque la sortie binaire est éteinte. Cet état est modifiable à partir d'une console *BACnet*, **mais on ne doit pas perdre de vue qu'une requête pour changer l'état d'une sortie binaire s'applique ou non selon l'état courant de la table des priorités (*« Priority_Array »*) dans le KC601**; se référer à l'*Annexe 1. Tables des priorités pour les sorties binaires* à la fin de ce manuel.

- *« Description »* (O/W) : texte modifiable pour décrire la sortie binaire. Les valeurs par défauts sont (se référer à la *Figure 20*) :
  - pour un groupe : *"Groupe No.X du KC401 No.Y"* où *X* est le numéro du groupe (valeur entre 1 et 4) et *Y* est le numéro de KC401 (valeur entre 1 et 4) sur lequel se retrouve le groupe;
  - le même principe s'applique pour les noms de sorties individuelles, selon le modèle : *"Sortie No.XX du KC401 No.Y"*, le numéro XX (valeur entre 01 et 16) débutant par un zéro pour les chiffres inférieurs à 10.
- *« Status_Flags »* (R) : il s'agit d'un regroupement de 4 *« drapeaux » (« flags »)* servant à signaler un problème ou un comportement particulier. Le KC601 utilise 2 de ces drapeaux :
  - *FAULT :* indique qu'une sortie binaire ne répond pas aux requêtes du KC601.
    - Cependant, la version actuelle du KC401 ne signale pas les sorties défectueuses ou pour lesquelles il n'y a pas de relais; dans un cas comme dans l'autre, une requête pour activer la sortie retourne alors un état à « OFF » sans signaler d'erreur.
    - Lorsqu'il y a faute, c'est que le KC401 ne répond plus. Toutes les sorties binaires tombent alors simultanément en faute.
  - *OUT_OF_SERVICE :* indique que le KC401 correspondant est en mode *"PROGRAM"*, pour la création de groupes. À noter que si un KC401 était « en faute », la communication étant rompue, il ne pourra apparaître *« OUT_OF_SERVICE ».*
- *« Event_State »* (R) : cette propriété retourne *« FAULT »* (se référer à l'item précédent) lorsqu'il y a une perte de communication avec le contrôleur KC401, mais retourne *« NORMAL »* si le contrôleur KC401 répond aux requêtes.
- *« Reliability »* (O) : cette propriété retourne *« NO_OUTPUT »* lorsqu'il y a une perte de communication avec contrôleur KC401, mais retourne *« NO_FAULT_DETECTED»* si le KC401 répond aux requêtes.
- *« Out_Of_Service »* (R) : cette propriété retourne *« TRUE »* pour chacune des sorties binaires lorsque le KC401 correspondant est en mode *« PROGRAM »* (les requêtes *BACnet* destinées aux KC401 étant alors accumulées dans le KC601, en attente d'être appliquées lors du retour au mode *« RUN »*), et *« FALSE »* dans le cas contraire.
- *« Polarity »* (R) : comme la polarité n'a jamais besoin d'être inversée pour les sorties binaires du KC601, la valeur retournée sera toujours : *« NORMAL ».*
- *« Priority_Array »* (R) : pour chaque sortie binaire, le KC601 gère un table de 16 priorités, bien qu'il rejette la priorité 6 ; pour plus de détails, se référer à l'*Annexe 1. Tables des priorités pour les sorties binaires* à la fin de ce manuel.
- *« Relinquish_Default »* (R) : valeur par défaut utilisée comme *« Present_Value »* lorsque la dernière priorité utilisée pour une sortie binaire est « libérée » (*« relinquished »*). Toute la table de priorité correspondante se retrouve alors composée uniquement de valeurs nulles et le niveau à appliquer à la sortie est celui pré-défini pour le paramètre : *« Relinquish_Default »*, fixé à *« BINARY_INACTIVE »* dans le KC601. Se référer à l'*Annexe 1. Tables des priorités pour les sorties binaires.*

### 5.1.3 « Analog Value Object Type »

C'est sous ce type d'objet que l'on peut connaître les options associées aux groupes et appliquées aux sorties individuelles. Pour visualiser ces options, il faut d'abord avoir configuré le KC601 pour qu'il retourne les *« valeurs analogiques »* ; se référer à la section : *3.1.2.2 Sélections des objets BACnet du KC601 accessibles via une console BACnet.*

Pour voir les options associées aux groupes, il faut aussi avoir configuré le KC601 pour qu'il retourne les états des sorties binaires associées aux groupes. Il en va de même pour les sorties individuelles.

**Propriétés supportées :**

- *« Object_Identifier »* (R) : le numéro d'identification de chacun de ces objets *BACnet* intégrés au KC601. Ce numéro est fixe.
  - S'il y a des groupes, le premier groupe aura la valeur « 1 » comme *« Object_Identifier »*, le second groupe la valeur « 2 » et ainsi de suite. Les sorties individuelles auront les numéros subséquents, suivant le dernier numéro de groupe.
  - S'il n'y a pas de groupe, la première sortie individuelle aura la valeur « 1 » comme *« Object_Identifier »* et ainsi de suite.
- *« Object_Name »* (R) :
  - le KC601 attribue aux valeurs analogiques retournant les options des **groupes** des noms (caractères alphanumériques) immuables selon le modèle : *« PROPERT_GRP_XX »* où *XX* est le numéro du groupe (valeur entre 01 et 16) dans la passerelle, débutant par un zéro pour les chiffres inférieurs à 10;
  - le même principe s'applique pour les noms de **sorties individuelles**, selon le modèle : *" PROPERT_OUT_XX"*, où *XX* est le numéro de la sortie (valeur entre 01 et 64).

| | Variable Name | Description |
|---|---|---|
| 1 | PROPERT_GRP_01 | Groupe_01= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| 2 | PROPERT_GRP_02 | Groupe_02= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| 3 | PROPERT_GRP_03 | Groupe_03= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| 4 | PROPERT_GRP_04 | Groupe_04= 1000:OFF, 100:ON, 10:TOE, 1:WARN |
| 5 | PROPERT_OUT_01 | Sortie No.01 = 10:TOE, 01:WARNING |
| 6 | PROPERT_OUT_02 | Sortie No.02 = 10:TOE, 01:WARNING |
| 7 | PROPERT_OUT_03 | Sortie No.03 = 10:TOE, 01:WARNING |
| 8 | PROPERT_OUT_04 | Sortie No.04 = 10:TOE, 01:WARNING |

*Figure 21 : Exemple d'affichage des noms générés par le KC601 (en français) pour ses objets de type « Analog Values » (avec groupes et sorties individuelles)*

- *« Object_Type »* (R) : le type d'objet BACnet = *« ANALOG_VALUE ».*
- *« Present_Value »* (R) : la valeur « réelle », avec des décimales *(« floating point »)* nulles, représente en fait l'état d'un champ de bits, chaque bit représentant une option (voir la *« Description »* qui suit, de même que la section : *4.4 Options pour les sorties des KC401*).
- *« Description »* (O) : texte non modifiable pour décrire comment est réparti le champ de bits pour représenter les options (se référer à la *Figure 21*) :
  - pour une valeur analogique associée à un groupe, le texte descriptif est :
    *"Group_XX= 1000:OFF, 100:ON, 10:TOE, 1:WARN"* où *XX* est le numéro du groupe (valeur entre 01 et 16) dans le KC601;
  - pour une valeur analogique associée à une sortie individuelle, le texte descriptif est : *"Sortie No.XX = 10:TOE, 01:WARNING"* où *XX* le numéro de la sortie (valeur entre 01 et 64) dans le KC601. À remarquer que seulement 2 options s'appliquent aux sorties individuelles.

- *« Status_Flags »* (R) : il s'agit d'un regroupement de 4 *« drapeaux » (« flags »)* servant à signaler un problème ou un comportement particulier. Le KC601 utilise 2 de ces drapeaux (se référer à la section *5.1.2 « Binary Output Object Type »*) :
  - *FAULT :* indique que le KC401, auquel s'applique une valeur analogique, ne répond plus aux requêtes du KC601.
  - *OUT_OF_SERVICE :* indique que le KC401 correspondant est en mode *"PROGRAM"*, pour la création de groupes.
- *« Event_State »* (R) : cette propriété retourne *« FAULT »* lorsqu'il y a une perte de communication avec le KC401 correspondant, mais retourne *« NORMAL »* si le KC401 répond aux requêtes.
- *« Reliability »* (O) : cette propriété retourne *« NO_OUTPUT »* lorsqu'il y a une perte de communication avec le KC401 correspondant, mais retourne *« NO_FAULT_DETECTED»* si le KC401 répond aux requêtes.
- *« Out_Of_Service »* (R) : cette propriété retourne *« TRUE »* lorsque le KC401 correspondant est en mode *« PROGRAM »* et *« FALSE »* lorsque le KC401 est de retour au mode *« RUN »*.
- *« Units »* : comme la valeur retournée pour *« Present_Value »* est un champ de bits, la valeur retournée par cette propriété est : *« NO_UNITS »*.

## 5.2 Commandes/Services BACnet supportés (selon les *modèle BIBBs / BACnet Interoperability Building Blocks* – Annexe K de la norme *ASHRAE*)

- *« Data Sharing-ReadProperty-B (DS-RP-B) » :* le KC601 répond aux requêtes de lecture d'une propriété à la fois *(ReadProperty)*. Pour les propriétés non supportées, un message d'erreur est retourné pour en aviser le requérant.
- *« Data Sharing-ReadPropertyMultiple-B (DS-RPM-B) »* : le KC601 répond aux requêtes de lecture de plusieurs propriétés demandées simultanément *(ReadPropertyMultiple)*. Encore ici, un message d'erreur avise le requérant des propriétés qui ne sont pas supportées.
- *« Data Sharing-WriteProperty-B (DS-WP-B) » :* le KC601 accepte les requêtes pour écrire une propriété à la fois. Pour les propriétés qui ne permettent pas l'écriture ou qui ne sont pas supportées, un message d'erreur est retourné pour indiquer que l'accès est refusé.
- *«Device Management-Dynamic Device Binding-B (DM-DDB-B) » :* le KC601 accepte les requêtes *« Who-Is »* et si le numéro d'identification *BACnet* du KC601 appartient à l'étendue recherchée, passée en paramètres avec la requête, le KC601 initie une réponse : *« I-Am »* avec ses coordonnées, mais seulement lorsqu'on lui passe le jeton *(« token »)*.
- *«Device Management-DeviceCommunicationControl-B (DM-DCC-B) »* le KC601 exécute les requêtes de : *« DeviceCommunicationControl »* en autant que le mot de passe fourni avec la requête corresponde à celui établi avec la console de configuration *Windows* (s'il y en a un). Selon les paramètres passés, les requêtes acceptées peuvent être de 3 natures :
  - *« DISABLE » :* pour désactiver la communication pour un certain nombre de minutes ou une période indéterminée si le paramètre correspondant passé avec la commande est nul.
  - *« DISABLE_INITIATION » :* pour empêcher le KC601 d'initier des requêtes pour un certain nombre de minutes ou pour une période indéterminée. En tant que *« master node »*, on désactive alors l'acceptation du jeton puisque l'on ne pourra initier de requête pour le passer.
  - *« ENABLE » :* pour annuler l'application de l'une ou l'autre des requêtes précédentes.
- *«Device Management-ReinitializeDevice-B (DM-RD-B) » :* le KC601 exécute les requêtes de réinitialisation : *« ReinitializeDevice »* en autant que le mot de passe fourni avec la requête corresponde à celui établi avec la console de configuration *Windows* (s'il y en a un). Les requêtes acceptées peuvent être de 2 natures :

- *« Cold Start » :* réinitialisation complète de la carte, incluant les accumulateurs de commandes en attente et de tous les compteurs, qu'ils soient affectés à la gestion de la communication ou aux diagnostiques. Cette réinitialisation est perceptible sur la carte par les LED (se référer à la section : *2.1.3 LED / Diodes Électroluminescentes*).
- *« Warm Start » :* réinitialisation uniquement de la communication *BACnet*, les commandes en attente destinées aux KC401 de même que les compteurs de diagnostiques n'étant pas affectés. Cette réinitialisation n'est pas détectable en examinant la carte…

## 5.3 *BACnet* Protocol Implementation Conformance Statement (PICS)

Tout appareil qui peut être intégré dans un environnement *BACnet* doit être décrit selon un formulaire de *« conformité d'implémentation du protocole BACnet »*, appelé *« PICS »*. Ce document identifie le mode de fonctionnement et les caractéristiques d'un appareil *BACnet* selon une un format et une terminologie particulière décrite dans la norme *BACnet*, entre autres :

- À quel profil *BACnet* normalisé notre appareil correspond t'il ?
- Une description de l'appareil, incluant ses fonctionnalités et l'identité de son fabriquant.
- Les blocs fonctionnels d'interopérabilité supportés dans notre appareil ; se référer dans ce manuel à la section *5.2 Commandes/Services BACnet supportés (selon les modèle BIBBs / BACnet Interoperability Building Blocks – Annexe K de la norme ASHRAE).*
- La liste des objets *BACnet* que l'on retrouve sur notre appareil, de même que leurs propriétés.
- Le type de communication supportée et les paramètres qui s'y rapportent.

Le *« PICS » (#121-38522)* du KC601 est complémentaire au manuel d'opération. Si on ne vous l'a pas déjà fourni avec ce manuel, vous n'avez qu'à en faire la demande auprès de votre représentant *Gentec/Kameleon*.

Pour davantage d'information sur en quoi consiste un *« PICS »*, se référer au manuel de la norme *BACnet* : *« ASHRAE Standard »*, à la section 22 : *« Conformance And Interoperability »* de même qu'à l'annexe « L » sur les descriptions et les profils des objets *BACnet* normalisés *("Descriptions and Profiles of Standardized BACnet Devices").*

# ANNEXES

## Annexe 1. Tables des priorités pour les sorties binaires

(se référer à la norme *BACnet*, section 19.2 : *Command Prioritization* )

### Annexe 1.a Les tables de priorité selon la norme ASHRAE

Si un appareil *« BACnet (device) »* est composé de *« sorties binaires »*, on doit pouvoir lui envoyer une commande d'écriture pour changer son état via la propriété *« Present_Value »*. Cette propriété doit avoir la caractéristique d'être *« commandable »* de sorte qu'une requête pour changer son état doit être composée (entre autre) de 2 paramètres : l'état désiré et une **priorité**. Ce système a été implanté dans la norme *ASHRAE* pour tenir compte que plusieurs *« entités commandantes » (« commanding entity »*, soit une application/console *BACnet*, un opérateur, etc.) peuvent envoyer des requêtes à une même sortie afin de changer son état. Pour éviter des conflits entre ces requêtes, une priorité est attribuée à chacune des *« entités commandantes » BACnet* qui peuvent transmettre une telle requête. Mais une console *BACnet* (par exemple) n'est pas limitée à un niveau de priorité pré-établi et peut transmettre des requêtes de diverses priorités, ce qui est cependant déconseillé dans la norme *ASHRAE* (section 19.2.1.3)

- L'**état** transmis à une sortie binaire est une valeur de type *« BACNET_BINARY_PV »*. Les 3 états permis pour ce type sont :
  - BINARY_INACTIVE = 0 ➔ OFF
  - BINARY_ACTIVE = 1 ➔ ON
  - BINARY_NULL = 2 ➔ NULL
- La **priorité** est une valeur entre 1 et 16, la priorité *« 1 »* étant la plus élevée (i.e. elle est appliquée avant toutes les autres priorités) alors que la priorité *« 16 »* est la plus négligeable.
- Certaines priorités sont déjà pré-définies :
  - **Priorité 1 :** Protection vitale manuelle (traduction libre de : *« Manual-Life Safety »*) ;
  - **Priorité 2 :** Protection vitale automatique (traduction libre de : *« Automatic-Life Safety »*) ;
  - **Priorité 5 :** Contrôle d'équipement critique (*« Critical Equipment Control »*) ;
  - **Priorité 6 :** Durée minimum *« On/Off »* (*« Minimum On/Off »*) ; cette priorité ne peut être utilisée que si on applique l'algorithme relié aux propriétés : *« Minimum_On_Time »* et *« Minimum_Off_Time »* (se référer à la *section 19.2.3* de la norme *ASHRAE*).
  - **Priorité 8 :** Opérateur manuel (*« Manual Operator »*).
- Dans un *« BACnet device »* composé de *« sortie(s) binaire(s)»*, il y a une ***« table de priorités »*** (*« Priority_Array »*) pour chacune des sorties binaires. Chaque table a donc 2 dimensions pour associer un état à chacune des 16 priorités :

| Priorités (#) | État (*On/Off/Null*) | Etat équivalent *BACnet* |
|---|---|---|
| Priorité 1 | NULL | BINARY_NULL |
| Priorité 2 | NULL | BINARY_NULL |
| Priorité 3 | NULL | BINARY_NULL |
| Priorité 4 | NULL | BINARY_NULL |
| Priorité 5 | **OFF** | **BINARY_INACTIVE** |
| Priorité 6 | NULL | BINARY_NULL |
| Priorité 7 | NULL | BINARY_NULL |
| Priorité 8 | **ON** | **BINARY_ACTIVE** |
| Priorité 9 | NULL | BINARY_NULL |
| Priorité 10 | NULL | BINARY_NULL |
| Priorité 11 | NULL | BINARY_NULL |
| Priorité 12 | NULL | BINARY_NULL |
| Priorité 13 | NULL | BINARY_NULL |
| Priorité 14 | NULL | BINARY_NULL |
| Priorité 15 | NULL | BINARY_NULL |
| Priorité 16 | NULL | BINARY_NULL |

*Figure 22 : Exemple d'une table de priorités, avec 2 priorités effectives (5 et 8).*

- Au départ, les 16 priorités de cette table ont leur valeur d'état à *« BINARY_NULL »* :
- Si une sortie binaire reçoit une requête *BACnet* par exemple de priorité 8 pour mettre la sortie à *« ON »*, alors la valeur *« BINARY_ACTIVE »* sera assignée pour la priorité 8 dans la table de la sortie visée. Comme parmi les 16 priorités, seule celle à 8 a alors une valeur différente de *« BINARY_NULL »*, l'état (*« ON »*) de cette priorité sera appliqué à la sortie.
- Si la même sortie reçoit ensuite une requête de priorité 5 pour mettre la sortie à *« OFF »*, on se retrouve alors avec 2 états non nuls dans la table de priorité de cette sortie. L'état de la priorité la plus élevée (5) étant alors *« BINARY_INACTIVE »*, la sortie correspondante sera mise à *« OFF »* (voir la *Figure 22*).
- Si la sortie reçoit une requête de priorité 7 pour mettre la sortie à *« ON »*, comme cette priorité est inférieure à la priorité effective la plus élevée (5), l'état *« BINARY_ACTIVE »* sera assignée dans la table à la priorité 7 sans être appliqué à la sortie :

| Priorités (#) | État (*On/Off/Null*) | Etat équivalent *BACnet* |
|---|---|---|
| Priorité 1 | NULL | BINARY_NULL |
| Priorité 2 | NULL | BINARY_NULL |
| Priorité 3 | NULL | BINARY_NULL |
| Priorité 4 | NULL | BINARY_NULL |
| Priorité 5 | **OFF** | **BINARY_INACTIVE** |
| Priorité 6 | NULL | BINARY_NULL |
| Priorité 7 | **ON** | **BINARY_ACTIVE** |
| Priorité 8 | **ON** | **BINARY_ACTIVE** |
| Priorité 9 | NULL | BINARY_NULL |
| Priorité 10 | NULL | BINARY_NULL |
| Priorité 11 | NULL | BINARY_NULL |
| Priorité 12 | NULL | BINARY_NULL |
| Priorité 13 | NULL | BINARY_NULL |
| Priorité 14 | NULL | BINARY_NULL |
| Priorité 15 | NULL | BINARY_NULL |
| Priorité 16 | NULL | BINARY_NULL |

*Figure 23 : Exemple d'une table de priorités, avec 3 priorités effectives (5, 7 et 8).*

- Si l'état passé avec la commande d'écriture est : *« BINARY_NULL »*, alors il s'agit d'une commande pour libérer (*« relinquish »*) la priorité passée en paramètre.
- Ainsi, si notre sortie reçoit une telle requête de priorité 5, alors la valeur mise dans la table pour cette priorité est : *« BINARY_NULL »*. On se retrouve alors avec des états non nuls pour les priorités 7 et 8. Comme la priorité 7 devient alors la plus haute priorité effective, son état est appliqué et la sortie passe alors de *« OFF »* à *« ON »*.

| Priorités (#) | État (*On/Off/Null*) | Etat équivalent *BACnet* |
|---|---|---|
| Priorité 1 | NULL | BINARY_NULL |
| Priorité 2 | NULL | BINARY_NULL |
| Priorité 3 | NULL | BINARY_NULL |
| Priorité 4 | NULL | BINARY_NULL |
| Priorité 5 | NULL | BINARY_NULL |
| Priorité 6 | NULL | BINARY_NULL |
| Priorité 7 | **ON** | **BINARY_ACTIVE** |
| Priorité 8 | **ON** | **BINARY_ACTIVE** |
| Priorité 9 | NULL | BINARY_NULL |
| Priorité 10 | NULL | BINARY_NULL |
| Priorité 11 | NULL | BINARY_NULL |
| Priorité 12 | NULL | BINARY_NULL |
| Priorité 13 | NULL | BINARY_NULL |
| Priorité 14 | NULL | BINARY_NULL |
| Priorité 15 | NULL | BINARY_NULL |
| Priorité 16 | NULL | BINARY_NULL |

*Figure 24 : État de la table de priorités précédente après avoir libéré (« relinquish ») la priorité 5.*

- Si la sortie reçoit une requête pour libérer (*« relinquish »*) la priorité 7, alors seule la priorité 8 demeure avec une valeur non nulle et son état *« ON »* est alors appliqué.
- Finalement, la transmission à notre sortie d'une requête pour libérer (*« relinquish »*) la dernière priorité effective (8) provoquera l'application de l'état qui a été défini par défaut via le paramètre : *Relinquish_Default.*

**Annexe 1.b  Les tables de priorités dans le contrôleur KC601**

- Toutes les sorties individuelles (pour gérer des relais contrôlant des lumières) et tous les groupes de sorties disponibles sur le KC601 sont représentés dans l'environnement *BACnet* comme des *« sorties binaires »*, en autant que vous ayez choisi dans la console de configuration de pouvoir les visualiser (se référer à la *Figure 25* de même qu'à la section : *3.1.2.2  Sélections des objets BACnet du KC601 accessibles via une console BACnet*).
- Si on a configuré le KC601 pour qu'il affiche toutes les sorties binaires :

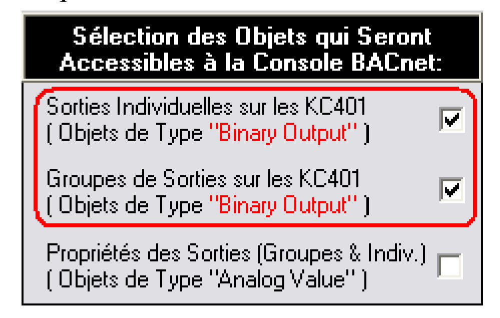



*Figure 25 : Sélection de tous les objets BACnet de type « Binary outputs » à partir du configurateur du KC601.*

alors chaque « sortie binaire » associée à un groupe contrôle un ensemble de « sorties binaires » associées aux sorties individuelles. Avec une requête *BACnet* pour mettre un groupe à *« ON »* ou à *« OFF »*, les tables de priorités des sorties individuelles seront affectées différemment selon l'option choisie pour le paramètre de configuration :




*Figure 26 : Choix de la manière dont les tables de priorités seront affectées lors du changement d'état d'un interrupteur mural.*

Se référer à la section *3.1.2.4  Autres paramètres ajustables pour les échanges BACnet* pour plus de détails.

- Dans une table de priorités, lorsque la dernière priorité non nulle est libérée (*« relinquished »*), l'état alors appliqué est la valeur par défaut établie pour : *« Relinquish_Default »*. Dans le KC601, cet état a été fixé à *« OFF »*.
- Lors de la réinitialisation du KC601, toutes les tables de priorités sont initialisées à : *« BINARY_NULL »*. Cependant, un état par défaut n'est pas appliqué aux sorties lors de cette réinitialisation; l'état courant des sorties est plutôt lu et retourné à la console *BACnet* sans associer l'état lu à une priorité. Si nous n'avions pas fonctionné de la sorte, il y aurait eu une réinitialisation de l'état des luminaires avec chaque démarrage ou réinitialisation du KC601, ce qui serait peu pratique dans un édifice, particulièrement pendant les heures de travail…
- Toutes les requêtes de **priorité 6** sont rejetées : le KC601 ne supporte pas les paramètres *« Minimum_Off_Time » / « Minimum_On_Time »* servant à conserver la sortie à *« Off » / « On »* durant une période minimum après un changement d'état à *« Off » / « On »,* selon un algorithme particulier. Or la norme *ASHRAE (section 19.2.3)* exige que l'on rejette la priorité 6 lorsque l'algorithme relié à ces 2 paramètres sur les durées minimum *« On/Off »* ne s'applique pas.



PAGE BLANCHE